

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ» ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

«До захисту допущено»

Завідувач кафедри

О.В. Коваль

(підпис) (ініціали, прізвище)

“ ” 2019р.

## **Магістерська дисертація**

зі спеціальності - 122 Комп'ютерні науки

за спеціалізацією - Комп'ютерний моніторинг та геометричне моделювання  
процесів і систем

на тему

Пошуково-аналітична система підтримки процесів аналізу товарного ринку

Виконав: студент 6 курсу, групи ТР-81мп

Шклярський Назар Олександрович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник

доц. к.т.н. Кузьмініх В.О

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_

(підпис)

Київ – 2019

**Національний технічний університет України  
„Київський політехнічний інститут” імені Ігоря Сікорського**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 122 Комп'ютерні науки

за спеціалізацією - Комп'ютерний моніторинг та геометричне моделювання процесів і систем

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
Коваль О.В.  
(прізвище, ініціали) \_\_\_\_\_ (підпис)

« \_\_\_\_ » \_\_\_\_\_ 2019р.

**З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

\_\_\_\_\_  
Шклярському Назару Олександровичу  
(прізвище, ім'я, по батькові)

1. Тема дисертації пошуково-аналітична система підтримки процесів аналізу товарного ринку

Науковий керівник Кузьмініх Валерій Олександрович, доцент, доц. к.т.н.  
(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від " \_\_ " \_\_\_\_\_ 2019 р. №

-  
2. Строк подання студентом роботи

-  
3. Об'єкт дослідження: комп'ютерні інформаційні системи і технології розумного будинку

4. Предмет дослідження: системи інтелектуального аналізу даних

5. Перелік питань, які потрібно розробити

- проаналізувати готові програмні рішення;

– спроектувати архітектуру системи для пошуково-аналітичної системи підтримки процесів аналізу товарного ринку:

– розробити програмне забезпечення для пошуково-аналітичної системи підтримки процесів аналізу товарного ринку.

6. Орієнтований перелік ілюстративного матеріалу: архітектура системи, графічне представлення інтерфейсу, приклади роботи програмного модулю.

7. Орієнтований перелік публікацій \_\_\_\_\_

1 Міжнародна наукова інтернет-конференція на тему "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення" (випуск 44). (м. Київ, 12 грудня 2019 року).

8. Дата видачі завдання "10" вересня 2018р.

### Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Відмітки
1.	Отримання завдання	10.09.2018р.	
2.	Аналіз вимог завдання, вибір методів і засобів розв'язання поставленої задачі	01.10.2018р. — 31.01.2019р.	
3.	Підготовка матеріалів магістерської роботи	01.10.2018р. — 20.11.2019р	
4.	Підготовка публікацій	01.03.2018р. — 26.04.2019р.	
5.	Захист програмного продукту	24.10.2019р	
6.	Передзахист	20.11.2019р	
7.	Захист	16 .12.2019р	

Студент

\_\_\_\_\_  
(підпис)

Шклярський Н.О.

(прізвище та ініціали,)

Науковий керівник

\_\_\_\_\_  
(підпис)

Кузьмініх В.О.

(прізвище та ініціали,)

# РЕФЕРАТ

## на магістерську дисертацію

виконану на тему: «Моделювання процесів міжагентної взаємодії в мережах Smart Grid»

студентом: Швайка Дмитром Андрійовичем

Магістерська дисертація складається зі вступу, шести розділів, висновку, переліку посилань з 32 найменувань, 1 додаток, і містить 31 рисунок, 9 таблиць. Повний обсяг магістерської дисертації складає 66 сторінок, з яких перелік посилань займає 3 сторінки, додатки – 1 сторінка.

**Актуальність теми.** Автоматизація є важливим напрямком розвитку підприємств та практики управління бізнес-процесами. Вона забезпечує ефективність роботи завдяки програмно-апаратним системам, які збільшують швидкість виконання та зменшують виробничі затрати. Інтернет речей в свою чергу забезпечує величезні апаратні та програмні можливості для реалізації проектів незалежно від сфери застосування.

Тому актуальним є дослідження можливостей Smart Grid, зокрема інтернету речей, в поєднанні з хмарними сервісами, зокрема, як ефективного рішення для автоматизації процесів в сфері торгівлі, так і в якості уніфікованого рішення для автоматизації процесів в будь-якій галузі або “розумному будинку”.

**Метою дослідження** є створення програмного забезпечення для автоматизації у будинку та для керування іот пристроями з іOS пристроїв.

Для досягнення поставленої задачі були сформульовані наступні **завдання дослідження**, що визначили логіку дослідження та його структуру:

- проаналізувати протокол керування пристроями;
- спроектувати архітектуру системи для моделювання процесів міжагентної взаємодії в мережах Smart Grid;
- розробити програмне забезпечення для моделювання процесів міжагентної взаємодії в мережах Smart Grid.

**Об’єктом дослідження** є комп’ютерні інформаційні системи і технології розумного будинку.

**Предметом дослідження** є системи інтелектуального аналізу даних в умовах розумного будинку.

**Методи дослідження.** Розв’язання поставлених задач виконувались з використанням наступних методів:

- метод аналізу;
- метод системного аналізу;
- метод порівняння;
- метод проектування логічних структур даних;
- метод логічного узагальнення результатів.

**Наукова новизна одержаних результатів.** Найбільш суттєвими науковими результатами магістерської дисертації є:

- удосконалення методу розпізнавання активності на базі даних сенсорів у реальному часі;
- розробка модулю для взаємодії з пристроями, які підтримують Apple HomeKit;
- розробка програмної системи яка використовуючи розроблену модель здійснює керування пристроями за створеними сценаріями.

**Практичне значення одержаних результатів** полягає в можливості використання програмної системи для автоматизації розумних будинків а також ручному керуванні пристроями з сумісних з Apple HomeKit гаджетів.

**Ключові слова.** *МОДЕЛЮВАННЯ, APPLE HOMEKIT, АВТОМАТИЗАЦІЯ, РОЗУМНИЙ БУДИНОК.*

## **ABSTRACT**

**on master's thesis**

on topic: «Modeling interagency interaction processes in Smart Grid networks»

Student: Dmytro Shvaika

The master's thesis consists of an introduction, six sections, conclusion, a list of references of 32 names, 1 appendix, and contains 31 figures, 9 tables. The full volume

of the master's thesis is 66 pages, of which the list of links occupies 3 pages, the annexes - 1 page.

**Topicality.** Automation is an important area of enterprise development and business process management practices. It delivers performance through software and hardware systems that increase execution speed and reduce production costs. The Internet of Things, in turn, provides enormous hardware and software for project implementation, regardless of scope.

Therefore, it is relevant to explore the capabilities of Smart Grid, in particular the Internet of Things, in combination with cloud services, in particular, as an effective solution for automation of trade processes, and as a unified solution for automation of processes in any industry or "smart home".

**The aim of the research is** to create software for home automation and control of iot devices from iOS devices.

To achieve this task, the following **research objectives** were formulated, which defined the logic of the study and its structure:

- analyze the device management protocol;
- to design the system architecture for modeling the processes of interagency interaction in Smart Grid networks;
- to develop software for modeling interagency interaction processes on Smart Grid networks.

**The subject of the study is** computerized information systems and smart home technologies.

**The subject of the study is** data mining systems in a smart home.

Research methods. The tasks were solved using the following methods:

- method of analysis;
- system analysis method;
- method of comparison;
- method of designing logical data structures;
- method of logical generalization of results.

**Scientific novelty of the obtained results.** The most significant scientific results of the master's thesis are:

- improvement of the method of recognition of activity on the basis of sensors in real time;
- developing a module to interact with devices that support Apple HomeKit;
- development of a software system that, using the developed model, manages the devices in the created scripts.

**The practical value of the results is** the ability to use a software system to automate smart homes, as well as manually control devices from Apple HomeKit-compatible gadgets.

**Keywords.** MODELING, APPLE HOMEKIT, AUTOMATION, SMART HOME.

# ЗМІСТ

ВСТУП.....	9
1. ЗАДАЧА ПОШУКОВО АНАЛІТИЧНОЇ СИСТЕМИ ПІДТРИМКИ ПРОЦЕСІВ АНАЛІЗУ ТОВАРНОГО РИНКУ.....	11
2. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ.....	12
2.1. GOOGLE ADWORDS:KEYWORD PLANNER.....	12
2.2. Wordstat Yandex.....	14
2.3. Google Trends.....	15
3. ЗАСОБИ ТА МЕТОДИ РЕАЛІЗАЦІЇ ВЕБ-СЕРЕДОВИЩА ДЛЯ ПОШУКОВО АНАЛІТИЧНОЇ СИСТЕМИ ПІДТРИМКИ ПРОЦЕСІВ АНАЛІЗУ ТОВАРНОГО РИНКУ .....	17
3.1. Мова програмування Python.....	17
3.2. Фреймворк Django.....	18
3.3. База даних.....	22
3.4. Середовище програмування.....	24
3.5. Розробка інтерфейсу користувача.....	26
4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ТА МАКЕТУ СИСТЕМИ.....	29
4.1. Структура бази даних.....	29
4.2. Клієнтський веб-інтерфейс.....	31
4.3. Структура модуля авторизації користувача.....	33
4.4. Модуль пошуку товарів по заданим критеріям.....	36
4.5. Обробка результатів пошуку та відбір товарів.....	41
4.6. Модуль візуалізації інформації.....	42
5. МЕТОДИКА РОБОТИ КОРИСТУВАЧА ІЗ ВЕБ-СЕРЕДОВИЩЕМ ДЛЯ ПОШУКОВО АНАЛІТИЧНОЇ СИСТЕМИ ПІДТРИМКИ ПРОЦЕСІВ АНАЛІЗУ ТОВАРНОГО РИНКУ .....	45
6. РОЗРОБКА СТАРТАП-ПРОЕКТУ.....	49
6.1. Опис ідеї проекту.....	49
6.2. Технологічний аудит ідеї проекту.....	50
6.3. Аналіз ринкових можливостей запуску стартап-проекту.....	51
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	57
Додаток 1.....	59



## ВСТУП

В 2019 порівняно з попереднім ринок електронної комерції виріс на третину. Українці купили товарів і послуг на суму 65 млрд грн.

Ринок інтернет товарів не дублює оффлайн ринок, який звичний для бізнесу. На цьому ринку якісні послуги інтернет маркетингу дозволять не тільки дізнатися про фактичний стан справ з продажами, ефективністю різних каналів просування, утриманням клієнтів і залученням нових, а й оцінити вибір, який інтернет ринок пропонує потенційним споживачам. Найголовніша відмінність оффлайн і інтернет ринку товарів і послуг – величезний вибір і практично відсутність будь-яких додаткових зусиль для потенційного покупця при пошуку альтернативи. Найчастіше аналіз ринку допомагає визначити нішу, яка поки що не надто зайнята та де можна отримати великий попит і дозволить оцінити попит з боку покупців, і вже на підставі отриманої інформації приймати відповідне рішення. Тому аналіз потрібно проводити ще на тому етапі, коли підприємець вибирає товар, який буде продавати в Інтернет-магазині. Все більше людей вбивають назву потрібного товару безпосередньо в пошук великого маркетплейса, минаючи "Гугл". У США 55% покупців відразу шукають товар на Amazon, Etsy або Ebay. В Україні частка прямого трафіку на Prom - 19%, Rozetka - 30%, OLX - 44%, Aliexpress - 50%, і з початку 2017 року вона зросла в середньому на 5% [1].

В рамках проекту розробляється програмний продукт, який дозволить проводити дослідження ринку роздрібної торгівлі, аналізувати попит і пропозиції та пропонувати найбільш рентабельні товари.

Перед власником онлайн магазину стоїть питання які товари пускати в реалізацію. За допомогою даного програмного продукту він може переглянути всі товари які є на ринку у великих маркетплейсах за потрібними категоріями. Подивитися якісну оцінку товару, яка дається на основі оцінки покупців і їх

коментарів. На основі інформації про попит і кількість постачальників дається оцінка про доцільність продажу цього товару.

Дана система може бути як основою для створення інтернет магазину так і використовуватися для вже працюючих. Загалом система дає власнику інструменти для контролю та аналізу закупок і продаж.

Для реалізації програмного забезпечення було обрано фреймворк Django та мову програмування JavaScript.

У першому розділі записки сформульовано мету й описано постановку задачі.

У другому розділі наводяться приклади схожих реалізованих систем.

У третьому розділі обґрунтовано вибір програмних засобів.

У четвертому розділі описується програмна реалізація.

У п'ятому розділі описано методику роботи користувача з програмною системою.

У шостому розділі описано розробку стартап проекту.

# **1. ЗАДАЧА РОЗРОБКИ ПОШУКОВО-АНАЛІТИЧНОЇ СИСТЕМИ ПІДТРИМКИ ПРОЦЕСІВ АНАЛІЗУ ТОВАРНОГО РИНКУ**

Метою розробки є реалізація веб-орієнтованого програмного засобу, що надає можливість автоматизувати роботу інтернет магазину.

Для досягнення мети необхідно:

- ❖ проаналізувати роботу існуючих систем;
- ❖ розширити теоретичні знання у відповідній тематиці;
- ❖ розробити модель системи;
- ❖ розробити графічний інтерфейс;
- ❖ здійснити програмну реалізацію системи.

Програмний додаток має включати наступний функціонал:

- ❖ пошук товарів по заданим критеріям;
- ❖ аналіз попиту на товари;
- ❖ знаходження схожих за характеристиками товарів;
- ❖ визначення найбільш оптимальних товарів для реалізації;
- ❖ візуалізація отриманих даних.

Основними користувачами програмної системи є:

- ❖ люди, які будуть використовувати дану систему для підбору товарів, які задовільняють їх потреби;
- ❖ підприємці, які прагнуть використовувати систему для роботи інтернет магазину.

## 2. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ

Щоб запропонувати те, що дійсно оцінять онлайн-покупці, потрібно зробити аналіз ринку й оцінити вигідність конкретної бізнес-ідеї. Вкрай важливо знати:

- ❖ хто, крім Вас, пропонує схожі товари;
- ❖ які їхні конкурентні переваги та сильні сторони;
- ❖ хто входить у поле цільової аудиторії покупців.

Найчастіше аналіз ринку допомагає визначити нішу, яка поки що не надто зайнята та де можна отримати великий попит. Тому аналіз потрібно проводити ще на тому етапі, коли підприємець вибирає товар, який буде продавати в Інтернет-магазині.

Є кілька зручних та недорогих способів досліджувати конкретну нішу на ринку. Вони дозволяють не витратити інвестиції даремно, й починати бізнес, будучи впевненими, що він окупиться, та ще й принесе прибуток.

### 2.1. GOOGLE ADWORDS: KEYWORD PLANNER

Планувальник ключових слів від Google відмінно підходить для таких цілей, як аналіз статистики за певними ключовими словами і оцінка трафіку.

Сервіс надає можливість:

- ❖ знайти ключові запити по фразі, сайту або категорії;
- ❖ підібрати ключі на основі вже наявних;
- ❖ проаналізувати статистику за певними словами;
- ❖ спрогнозувати, скільки показів і кліків буде по ключових запитах при певних ставках і бюджетах;
- ❖ отримати прогнози по існуючим кампаніям і запитам аккаунта.

Google Keyword Planner має 2 розділи, в одному з яких можна працювати з пошуком нових слів і збирати статистику запитів, а в другому - отримувати прогнози і планувати бюджет.

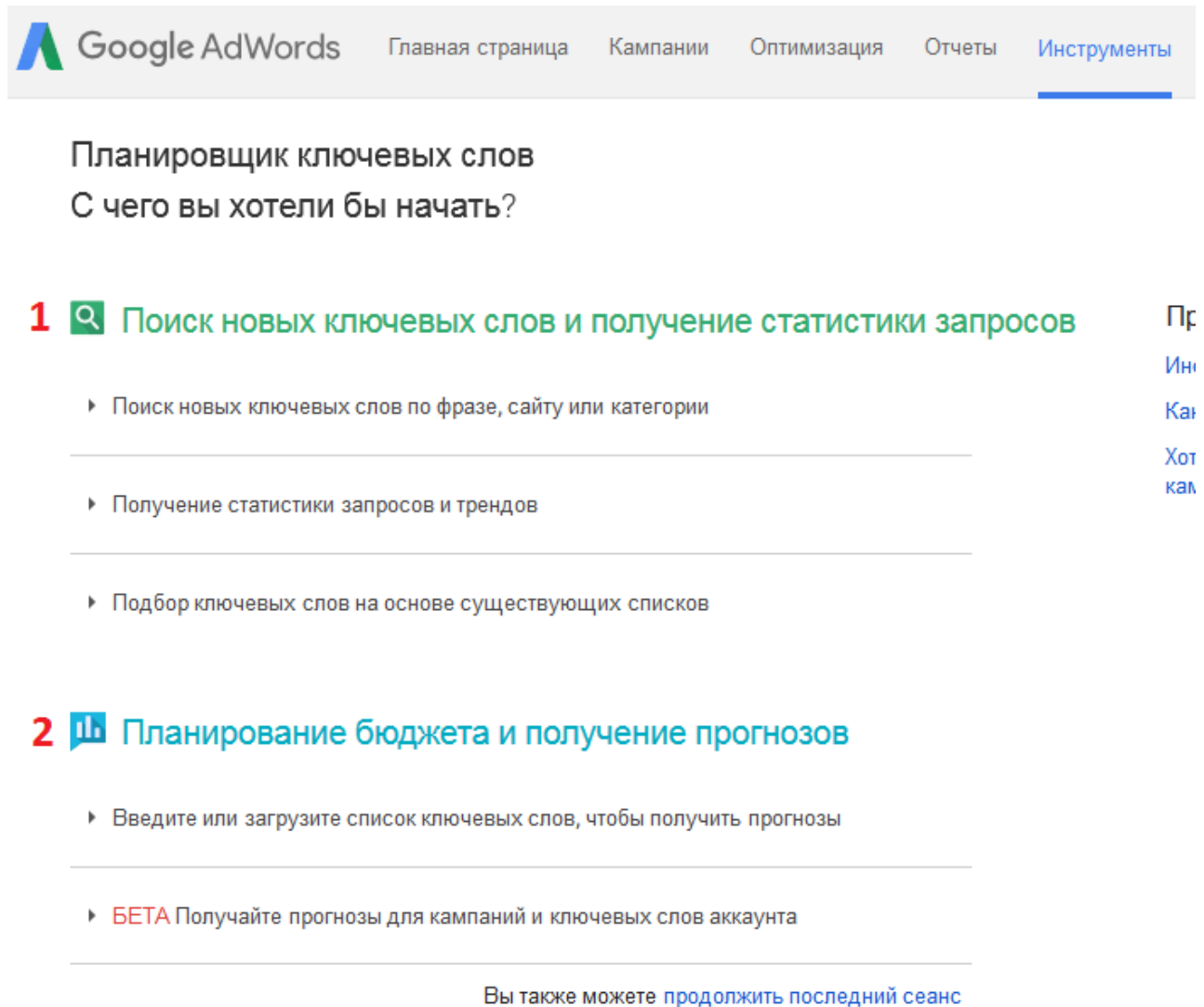


Рисунок 2.1 — Интерфейс сервису Google Adwords Keyword Planner

Keywords Planner - незамінний інструмент для підбору слів в «Гугл АдВордса». З його допомогою фахівці з контекстної реклами дізнаються, які ключові запити рекомендуються при створенні нових кампаній, і можуть підбирати нові ключі на основі вже наявних для розширення старих кампаній. Тут же фахівці планують бюджет і ставки. Активно користуються сервісом і SEO-фахівці.

Планувальник ключових слів допомагає знайти ключові запити і визначити їх частотність і конкурентність при складанні семантичного ядра сайту.

## 2.2. Wordstat Yandex

Wordstat Yandex - це безкоштовний сервіс Яндекс, призначений для оцінки користувальницького інтересу до різних тематик і підбору ключових слів для SEO-оптимізації і контекстної реклами. Крім того, за допомогою Wordstat Yandex можна оцінити сезонність і географічну залежність пошукових запитів.

Серед переваг сервісу слід виділити простоту і високу швидкість роботи. Цей безкоштовний сервіс має відмінний функціонал. З його допомогою можна:

- ❖ дізнатися частотність запиту в Яндексі в точній, фразовій і широкій відповідності;
- ❖ отримати статистику по тому чи іншому регіону;
- ❖ використовувати «Історію запитів» для визначення сезонності.

Тепер про мінуси. Просунуті SEO-фахівці сходяться на думці, що підбір ключових слів в Яндекс.Вордстат дозволяє скласти список ключів обсягом лише до 50% від повноцінного семантичного ядра. Для збереження даних на комп'ютер і пошуку інших фраз потрібно використання додаткових сервісів.

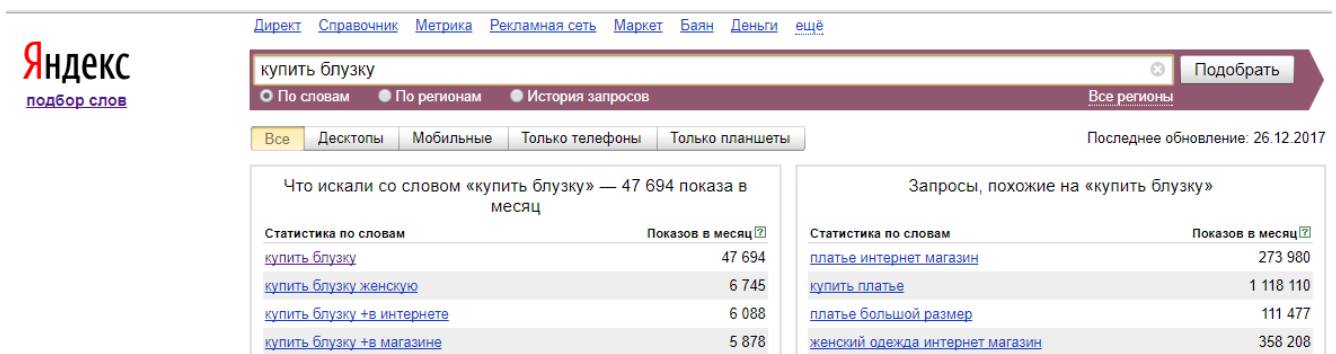


Рисунок 2.2 — Интерфейс сервісу ЯНДЕКС.WORDSTAT

Ще один недолік - WordStat показує тільки 2000 вкладені фрази (до 40 сторінок), упускаючи низькочастотні запити, які призводять додатковий трафік. Багатьох дратує часте введення «капчі», до того ж існує ймовірність блокування облікового запису.

## 2.3. Google Trends

Система Google пропонує зручне і просте засіб, що дозволяє вирішити цю проблему. Сервіс Google Trends дає можливість вести аналіз сезонності і популярності трендів пошукових запитів. Це зручний інструмент інтернет-маркетингу, який надає нам статистику частоти введення в пошуковий рядок Google запитів щодо необхідної нас тематиці.

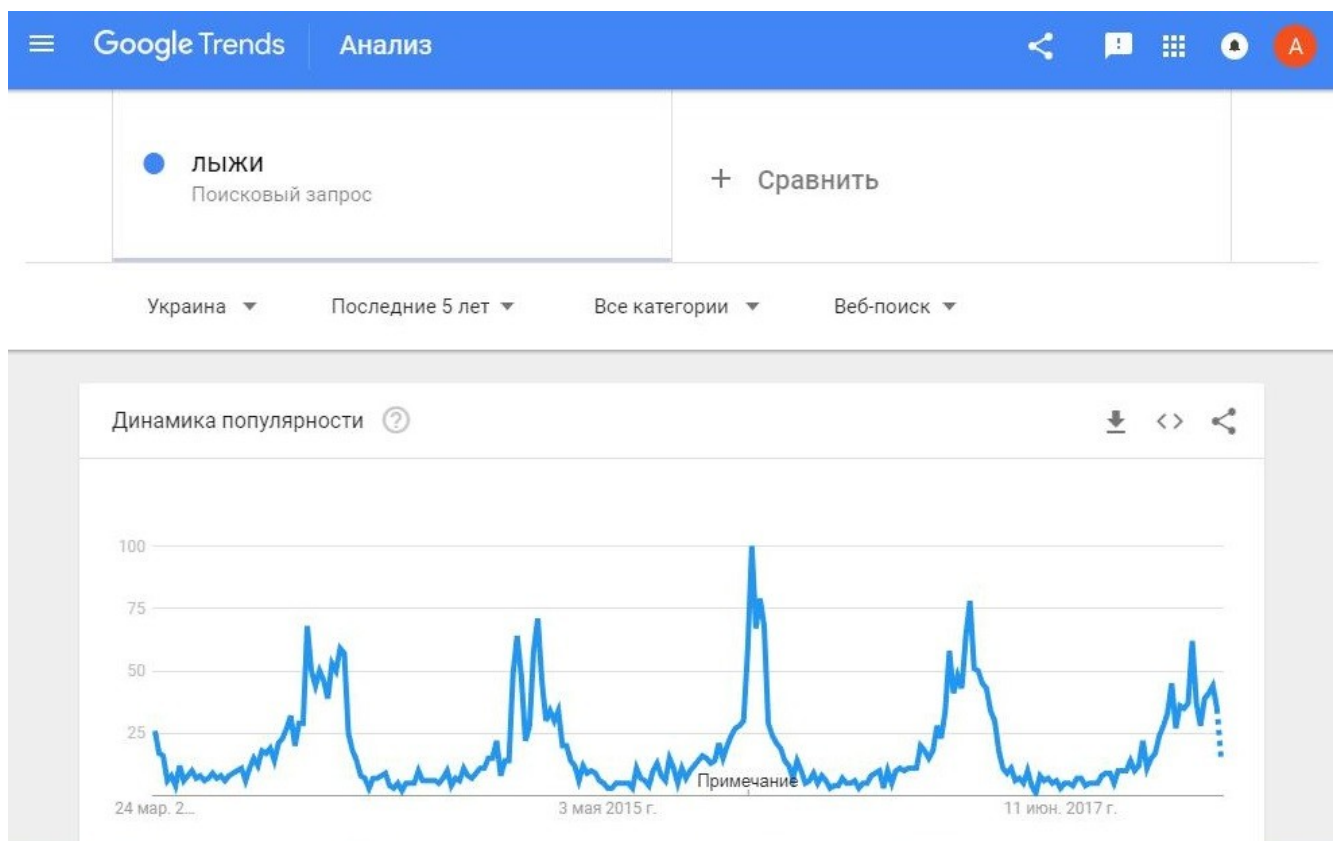


Рисунок 2.3 — Інтерфейс сервісу Google Trends

Одним з найважливіших моментів у процесі просування сайту, товару, послуги є відстеження тенденцій попиту. Як засіб статистичної вибірки, Google Trends дозволяє маркетологу:

- вести аналіз загального попиту по тематиці;
- визначати сезонність запитів по трендам в шуканій ніші;
- відслідковувати результативність анкорів;
- аналізувати популярність тренда в розрізі географії.

Google Trends не призначений для збору або аналізу частотності великої кількості пошукових запитів, тому деякі інтернет-маркетологи його ігнорують. Але для аналізу сезонності, поточних трендів або прогнозування попиту - це один з кращих безкоштовних інструментів.



### 3. ЗАСОБИ ТА МЕТОДИ РЕАЛІЗАЦІЇ

Одним із головних завдань при розробці програмного продукту є вибір таких засобів, які б полегшили роботу програміста, надавши всі необхідні інструменти для реалізації поставленого завдання.

При реалізації додатку для аналітичної системи було обрано наступні технології: Django, Python, MongoDB.

#### 3.1. Мова програмування Python

Python - універсальна мультипарадигменна скриптова мова програмування.

Python не призначений для чогось конкретного, це мова загального призначення: підходить для web-проектів, для desktop- і серверних додатків.

Пітон підтримує будь-який стиль програмування.

До 1960-х років було тільки дві групи мов: системні та командні. На системних мовах писали програмісти, а на командних - системні адміністратори. Згодом з'явилися скриптові мови - щось середнє між системними і командними.

Спочатку Python створювався з прицілом на об'єктноорієнтованість. Крім того, це дуже доброзичлива для вивчення і роботи мова, яка розроблялася так, щоб в ній було легко розібратися: максимальне приховування складнощів і дуже великі можливості. На Python можна писати будь-якого роду розширення, використовувати його в іграх як мову для штучного інтелекту, активно вбудовувати в інші додатки. Python вже став стандартом в проектуванні мов, і багато нових мов створюються з оглядкою на Python і використовують його конструкції. Мова Go, наприклад, «динамічна як Python і швидкий як C». Тим, хто вже працював з Python, потрібно мінімум зусиль, щоб освоїти ці мови.

### Основні переваги Python:

- ❖ Низький поріг входження: людині, знайомому з програмуванням, досить півгодини, щоб почати писати на ньому корисні для себе скріпти, а не знайомому - Python дозволяє легко відкрити для себе програмування і спробувати свої сили в ньому.
- 1. Добре спроектований: Python увібрав в себе сучасні тенденції в програмуванні «з нуля». Крім того, він динамічно розвивається: процес включення нових конструкцій в мову добре налагоджений, і він продовжує вбирати в себе прийоми функціонального програмування, аспектно-орієнтованого програмування.
- 2. Легко читається синтаксис (в порівнянні з C ++, Perl, PHP): дозволяє легко читати чужий код, розбиратися в давно написаному власному коді. У поєднанні зі сказаним вище це налаштовує творців бібліотек на простоту і логічність інтерфейсів.
- 3. Величезна кількість бібліотек з кодом на будь-який випадок життя: будь то робота з таблицями Excel, зображеннями або мережею Twitter.
- 4. Переносимість: Python реалізований під усіма поширеними операційними системами і на безлічі архітектур - Windows, Linux, MacOS, навіть на міні-комп'ютерах Arduino. Система залежностей добре продумана, і розгортання додатків на іншій машині відбувається легко і без сюрпризів.

## 3.2. Фреймворк Django

Django - це програмний каркас з багатими можливостями, що підходить для розробки складних сайтів і веб-додатків, написаний на мові програмування Python.

Django - фреймворк для веб-додатків на мові Python. Один з основних принципів фреймворка - DRY (do not repeat yourself). Веб-системи на Django будуються з одного або декількох додатків, які рекомендується робити відчужуваними і підключаємими. Це одне з помітних архітектурних відмінностей цього фреймворка від деяких інших (наприклад, Ruby on Rails). Також, на відміну

від багатьох інших фреймворків, обробники URL в Django конфігуруються явно (за допомогою регулярних виразів), а не автоматично задаються зі структури контролерів.

Django проектувався для роботи під управлінням Apache (з модулем `mod_python`) і з використанням PostgreSQL в якості бази даних. В даний час, крім PostgreSQL, Django може працювати з іншими СУБД: MySQL (MariaDB), SQLite, Microsoft SQL Server, DB2, Firebird, SQL Anywhere і Oracle. Для роботи з базою даних Django використовує власний ORM, в якому модель даних описується класами Python, і по ній генерується схема бази даних.

Архітектура Django схожа на «Модель-Представлення-Контролер» (MVC). Контролер класичної моделі MVC приблизно відповідає рівню, який в Django називається Представлення (View), а презентаційна логіка Представлення реалізується в Django рівнем Шаблонів (Templates). Через це рівневу архітектуру Django часто називають «Модель-Шаблон-Представлення» (MTV).

Спочатку розробка Django велася для забезпечення більш зручної роботи з новинними ресурсами, що досить сильно відбилося на архітектурі: фреймворк надає ряд засобів, які допомагають у швидкій розробці веб-сайтів інформаційного характеру. Наприклад, розробнику не потрібно створювати контролери та сторінки для адміністративної частини сайту, в Django є вбудований додаток для керування вмістом, яке можна включити в будь-який сайт, зроблений на Django, і яке може управляти відразу декількома сайтами на одному сервері. Адміністративне додаток дозволяє створювати, змінювати і видаляти будь-які об'єкти наповнення сайту, протоколюючи всі скоєні дії, і надає інтерфейс для управління користувачами і групами (з пооб'єктного призначенням прав).

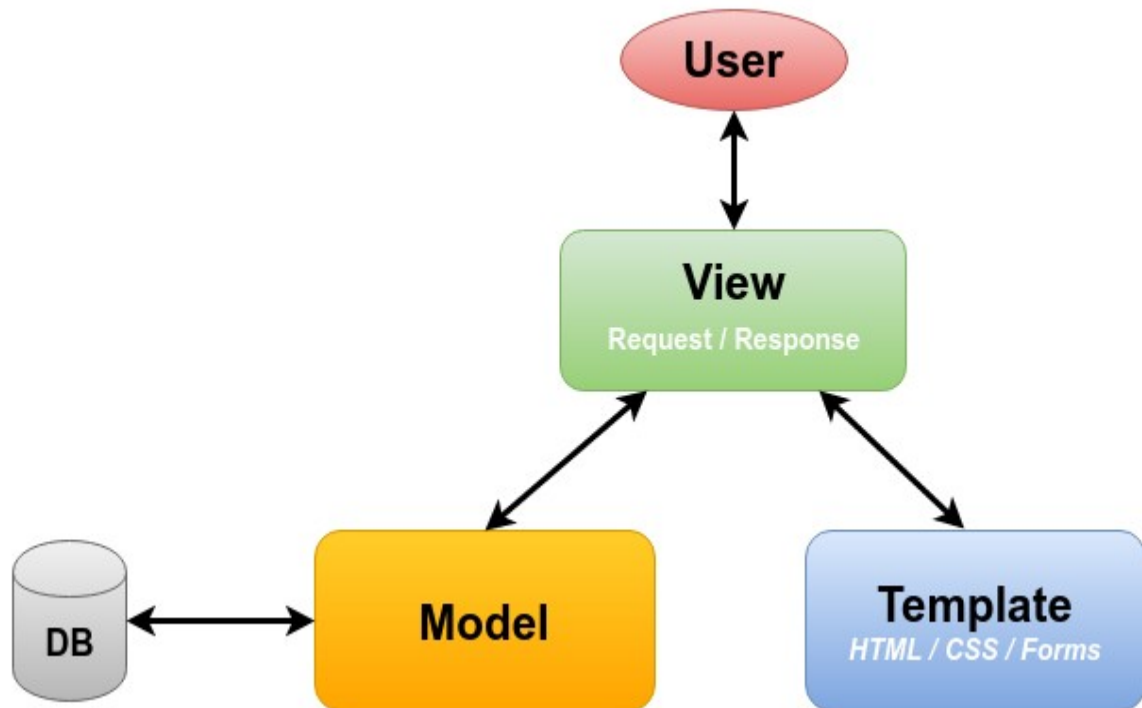


Рисунок 3.1 — Шаблон MTV

Веб-фреймворк Django використовується в таких великих і відомих сайтах, як Instagram, Disqus, Mozilla, The Washington Times, Pinterest, lamoda і ін.

Деякі можливості Django:

5. ORM, API доступу до БД з підтримкою транзакцій
6. вбудований інтерфейс адміністратора, з уже наявними перекладами багатьма мовами
7. диспетчер URL на основі регулярних виразів
8. розширювана система шаблонів з тегами і спадкуванням
9. система кешування
10. інтернаціоналізація
11. підключається архітектура додатків, які можна встановлювати на будь-які Django-сайти
12. «Generic views» - шаблони функцій контролерів
13. авторизація та аутентифікація, підключення зовнішніх модулів аутентифікації: LDAP, OpenID та інші.

14. система фільтрів ( «middleware») для побудови додаткових обробників запитів, як наприклад включені в дистрибутив фільтри для кешування, стиснення, нормалізації URL і підтримки анонімних сесій
15. бібліотека для роботи з формами (успадкування, побудова форм по існуючій моделі БД)
16. вбудована автоматична документація по тегам шаблонів і моделей даних, доступна через адміністративне додаток

Деякі компоненти фреймворка між собою пов'язані слабо, тому їх можна досить просто замінювати на аналогічні. Але з деякими (наприклад, з ORM) це зробити не дуже просто. Крім можливостей, вбудованих в ядро фреймворка, існують пакети, що розширюють його можливості.

На базі Django розроблено досить багато готових рішень, які розповсюджуються під вільною ліцензією, серед яких системи для управління інтернет-магазинами, універсальні системи управління вмістом, а також більш цілеспрямовані проекти.

### 3.3. База даних

MongoDB реалізує новий підхід до побудови баз даних, де немає таблиць, схем, запитів SQL, зовнішніх ключів і багатьох інших речей, які притаманні об'єктно-реляційних баз даних.

З часів динозаврів було звичайною справою зберігати всі дані в реляційних базах даних (MS SQL, MySQL, Oracle, PostgreSQL). При цьому було не так важливо, а чи підходять реляційні бази даних для зберігання даного типу даних чи ні.

На відміну від реляційних баз даних MongoDB пропонує документо-орієнтовану модель даних, завдяки чому MongoDB працює швидше, має кращу масштабованість, її легше використовувати.

Але, навіть враховуючи всі недоліки традиційних баз даних і гідності MongoDB, важливо розуміти, що завдання бувають різні і методи їх вирішення бувають різні. В якійсь ситуації MongoDB дійсно поліпшить продуктивність вашої програми, наприклад, якщо треба зберігати складні за структурою дані. В іншій же ситуації краще буде використовувати традиційні реляційні бази даних. Крім того, можна використовувати змішані підхід: зберігати один тип даних в MongoDB, а інший тип даних - в традиційних БД.

Вся система MongoDB може представляти не тільки одну базу даних, що знаходиться на одному фізичному сервері. Функціональність MongoDB дозволяє розташувати кілька баз даних на декількох фізичних серверах, і ці бази даних зможуть легко обмінюватися даними і зберігати цілісність.

Одним з популярних стандартів обміну даними та їх зберігання є JSON (JavaScript Object Notation). JSON ефективно описує складні за структурою дані. Спосіб зберігання даних в MongoDB в цьому плані схожий на JSON, хоча формально JSON не використовується. Для зберігання в MongoDB застосовується формат, який називається BSON (Бісон) або скорочення від binary JSON.

BSON дозволяє працювати з даними швидше: швидше виконується пошук і обробка. Хоча треба зазначити, що BSON на відміну від зберігання даних в форматі JSON має невеликий недолік: в цілому дані в JSON-форматі займають менше місця, ніж в форматі BSON, з іншого боку, даний недолік з лишком окупається швидкістю.

MongoDB написана на C ++, тому її легко перенести на найрізноманітніші платформи. MongoDB може бути розгорнута на платформах Windows, Linux, MacOS, Solaris. Можна також завантажити вихідний код і самому скомпілювати MongoDB, але рекомендується використовувати бібліотеки з офсайта.

Якщо реляційні бази даних зберігають рядки, то MongoDB зберігає документи. На відміну від рядків документи можуть зберігати складну за структурою інформацію. Документ можна уявити як сховище ключів і значень.

Однак при всіх відмінностях є одна особливість, яка зближує MongoDB і реляційні бази даних. У реляційних СУБД зустрічається таке поняття як первинний ключ. Це поняття описує якийсь стовпець, який має унікальні значення. У MongoDB для кожного документа є унікальний ідентифікатор, який називається `_id`. І якщо явно не вказати його значення, то MongoDB автоматично згенерує для нього значення.

Кожному ключу зіставляється певне значення. Але тут також треба враховувати одну особливість: якщо в реляційних базах є чітко окреслена структура, де є поля, і якщо якесь поле не має значення, йому (в залежності від налаштувань конкретної бд) можна привласнити значення NULL. У MongoDB все інакше. Якщо якомусь ключ не порівнювати значення, то цей ключ просто опускається в документі і не вживається.

Якщо в традиційному світі SQL є таблиці, то в світі MongoDB є колекції. І якщо в реляційних БД таблиці зберігають однотипні жорстко структуровані об'єкти, то в колекції можуть містити найрізноманітніші об'єкти, що мають різну структуру і різний набір властивостей.

Система зберігання даних в MongoDB представляє набір реплік. У цьому наборі є основний вузол, а також може бути набір вторинних вузлів. Всі вторинні вузли зберігають цілісність і автоматично оновлюються разом з оновленням головного вузла. І якщо основний вузол з якихось причин виходить з ладу, то один з вторинних вузлів стає головним.

Відсутність жорсткої схеми бази даних і в зв'язку з цим потреби при щонайменшій зміні концепції зберігання даних пересоздавать цю схему значно полегшують роботу з базами даних MongoDB і подальшим їх масштабуванням. Крім того, економиться час розробників. Їм більше не треба думати про пересозданні бази даних і витратити час на побудову складних запитів.

Однією з проблем при роботі з будь-якими системами баз даних є збереження даних великого розміру. Можна зберігати дані в файлах, використовуючи різні мови програмування. Деякі СУБД пропонують спеціальні типи даних для зберігання бінарних даних в БД (наприклад, BLOB в MySQL).

На відміну від реляційних СУБД MongoDB дозволяє зберігати різні документи з різним набором даних, однак при цьому розмір документа обмежується 16 мб. Але MongoDB пропонує рішення - спеціальну технологію GridFS, яка дозволяє зберігати дані за розміром більше, ніж 16 мб.

Система GridFS складається з двох колекцій. У першій колекції, яка називається files, зберігаються імена файлів, а також їх метадані, наприклад, розмір. А в іншій колекції, яка називається chunks, у вигляді невеликих сегментів зберігаються дані файлів, зазвичай сегментами по 256 кб.

Для тестування GridFS можна використовувати спеціальну утиліту mongofiles, яка йде в пакеті mongodb.

### **3.4. Середовище програмування**

Для написання застосунку було використано середовище програмування PyCharm.



Python став однією з найпопулярніших мов програмування в світі. PyCharm - одна з найбільш ефективних інтегрованих середовищ розробки. За допомогою інструментів цієї IDE розробники можуть швидко написати код Python і перевірити його на наявність помилок.

Тут є багато стандартних функцій, які ви знайдете на інших платформах IDE. Авто-відступ, підсвічування синтаксису і автозавершення коду - це лише деякі з них. PyCharm починає перевершувати, коли виходить за рамки цих основних функцій, щоб надати великий інструментарій для професійних розробників.

Для початку ця IDE практична. Код в PyCharm інтегрується з іншими платформами розробки, щоб спростити процес фактичного використання коду Python. Програма працює з Django, Flask, Pyramid і іншими популярними програмами. Підтримка платформи створює єдність, якого немає в багатьох інших середовищах розробки. Цей тип установки може заощадити розробникам час і гроші.

Також є інтеграція з IPython. Це дозволяє розробникам легко працювати над складними шматками коду. Проекти виконуватимуться набагато швидше, ніж без можливості обмінюватися фрагментами коду.

Програма також може бути налаштована відповідно до унікальними потребами розробника. Більше 50 плагінів доступні для безкоштовного скачування. Ці доповнення значно розширюють функціональність середовища. PyCharm також має інтуїтивно зрозумілу і інтерактивну консоль, яка буде стимулювати нових розробників, оскільки вони більше працюють з мовою. Вбудований термінал також надається.

Наявність IDE, яка буде тестувати і налагоджувати код, є обов'язковим, але ця програма пропонує набагато більше за дуже доступною ціною. План підписки з максимальним набором функцій починається менш ніж за 10 доларів в місяць, а також існує безкоштовна версія програмного забезпечення для тих, хто працює над проектами з відкритим вихідним кодом.

Переваги Pycharm:

- ❖ більше можливостей, ніж в аналогічних платформах IDE;
- ❖ інтегрується з декількома платформами;
- ❖ рефакторинг підтримки;
- ❖ удосконалений механізм автозаповнення.

Але також існують певні недоліки. PyCharm може повільно завантажуватися, а настройки за замовчуванням, можливо, доведеться підкоригувати для існуючих проектів.

### **3.5. Розробка інтерфейсу користувача**

Для розробки інтерфейсу користувача було вирішено обрати фреймворк Bootstrap.

Спочатку Bootstrap був створений розробниками Твіттера для внутрішнього користування, але через деякий час він був викладений у відкритий доступ і став зручним набором інструментів для розробки призначених для користувача інтерфейсів будь-якої складності.

До переваг Bootstrap можна віднести:

- ❖ скорочення часу, необхідного для створення макета сайту. Фреймворк пропонує достатню кількість шаблонів і готових рішень;
- ❖ кросбраузерну і адаптивність. Сайти, створені за допомогою Bootstrap, ідентично відображаються на різних пристроях і в сучасних браузерах;
- ❖ простота використання. Для роботи з фреймворком будуть потрібні елементарні навички верстки;
- ❖ легкість в освоєнні. Достатня кількість навчальних відеоматеріалів допоможе вам швидко і без особливих труднощів освоїти Bootstrap.

Багато тем для популярних систем управління вмістом (CMS) розроблені з використанням Bootstrap, що доводить рівень його якості. По суті він являє собою фреймворк, що містять компоненти CSS, HTML і JavaScript, а також власні стилі і шрифти.

1	1	1	1	1	1	1	1	1	1	1	1
2		2		2		2		2		2	
3			3			3			3		
4				4				4			
6						6					
12											

Рисунок 3.2 — Розмітка колонок в Bootstrap

Якщо говорити про адаптивних сайтах, що містять класичні елементи (шапку, футер, контент і бічну колонку), потрібно брати до уваги необхідність розрахунку ширини кожного з них. Зазвичай обчислення ведуться у відсотках, і якщо з шапкою і підвалом все зрозуміло (їх ширина зазвичай дорівнює 100%), то з іншими блоками виникають труднощі. На стаціонарних десктопах співвідношення контенту і бічний колонки може бути 75 до 25 або 80 до 20. При зменшенні вікна цей варіант неприпустимий, необхідно скидати до 100% і робити обтікання.

У таких ситуаціях необхідна сітка. Вона дає можливість задати класи об'єктів, вказавши при цьому їх ширину і характер відображення на різних пристроях. Сітка Bootstrap є таблицею, що містить ряди і стовпці. Ви можете створювати сітку всередині вже існуючої, тим самим позбавляючи себе від необхідності вручну вводити медіа-запити. Таким чином, значно спрощує і прискорює процес створення адаптивних блоків.

Крім сітки, фреймворк включає в себе безліч компонентів: кнопки, навігаційні елементи меню, форми, елементи оповіщення, таблиця, модальні вікна, спливаючі підказки. Деякі з цих елементів вам ніколи не знадобляться.

Але недоліки у даного фреймворка так само є. Їх всього два.

Перший. Шаблони, пропоновані Bootstrap, містять набагато більше інформації, ніж потрібно насправді. Коли розробник самостійно створює макет, він використовує тільки необхідний мінімум для реалізації задуманих ідей. Шаблон повинен враховувати безліч інших варіантів, тобто бути універсальним. Вирішити цю проблему можна, контролюючи компоненти, які ви будете переносити css-файл (по суті, ви можете використовувати тільки сітку, вводячи інші дані вручну).

Другий недолік полягає в шаблонності дизайну. Часто сайти, створені за допомогою фреймворка, схожі один на інший: однакова структура, кнопки, навігаційні елементи. Втрачається унікальність дизайну, що не завжди доречно. Цей недолік теж легко усунути. Намагайтеся мінімізувати використання готових рішень Bootstrap, змінюйте шаблон відповідно до індивідуального рішенням дизайнера і запитів замовника.

Bootstrap дозволяє створювати сайти з найрізноманітнішим дизайном і структурою, головне навчитися правильно використовувати всі інструменти, які пропонує даний фреймворк фронтенд-розробникам.

## 4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ТА МАКЕТУ СИСТЕМИ

Програмна реалізація додатку представляє собою п'ять модулів:

- ❖ клієнтський веб-інтерфейс;
- ❖ авторизація користувача;
- ❖ пошук товарів по заданим критеріям;
- ❖ обробка результатів пошуку та відбір товарів.
- ❖ візуалізація інформації.

Далі буде детально розглянуто кожен із складових компонентів системи.

### 4.1. Структура бази даних

Було розроблено сутності:

- ❖ User – зберігає інформацію про користувача;
  - id – id користувача;
  - password –пароль користувача;
  - last\_login – останній раз коли користувач був в системі;
  - username – логін користувача;
  - first\_name – ім'я користувача;
  - email – електронна пошта користувача;
  - is\_staff – чи є адміністратором;
  - is\_active – чи знаходиться в мережі;
  - date\_joined – дата створення користувача;
  - last\_name – прізвище користувача.
    - Product – зберігає інформацію про товар;
      - o id – id продукту;

- o product\_name – назва товару;
- o product\_links – посилання на товар;
- o product\_cost – ціна товару;
- o product\_photos – фото товару;
- o product\_val – кількість пропозицій на ринку;
- o product\_google – інформація про попит товару;
- o product\_charact – короткі характеристик;
- o product\_graph\_pop – графік популярності товару;
- o product\_graph\_usd – графік зміни ціни в доларах;
- o product\_graph\_uah – графік зміни ціни в гривнях;
- o product\_graph\_qual – графік зміни кількості пропозицій товару на ринку.

❖ Characteristics – зберігає детальні характеристики.

- o id– id запису;
- o characteristics\_id – id товару, якому належить характеристика;
- o characteristics\_name – назва характеристики;
- o characteristics\_name - значення характеристики.

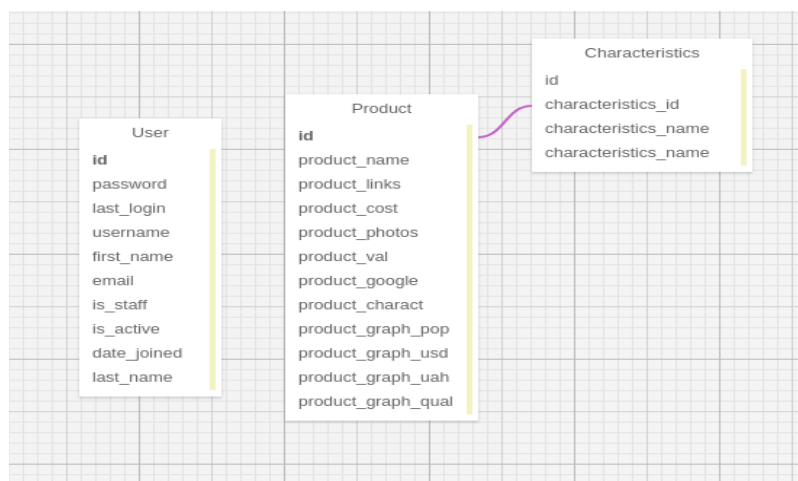


Рисунок 4.1 — Структура бази даних

## 4.2. Клієнтський веб-інтерфейс

У традиційному інформаційному веб-сайті, веб-додаток очікує запитів HTTP від веб-браузера (або іншого клієнта). Коли запит отримано, додаток розробляє то, що необхідно на основі URL-адреси і, можливо, інформації в POST або в GET запитах. Залежно від того, що потрібно, він може читати або записувати інформацію з бази даних або виконувати інші завдання, необхідні для задоволення запиту. Додаток потім поверне відповідь веб-браузеру, часто динамічно створюючи сторінку HTML для перегляду браузера, вставляючи отримані дані в HTML шаблон.

Веб-додатки написані на Django зазвичай групують код, який обробляє кожен з цих кроків, в окремі файли(рис 4.2):

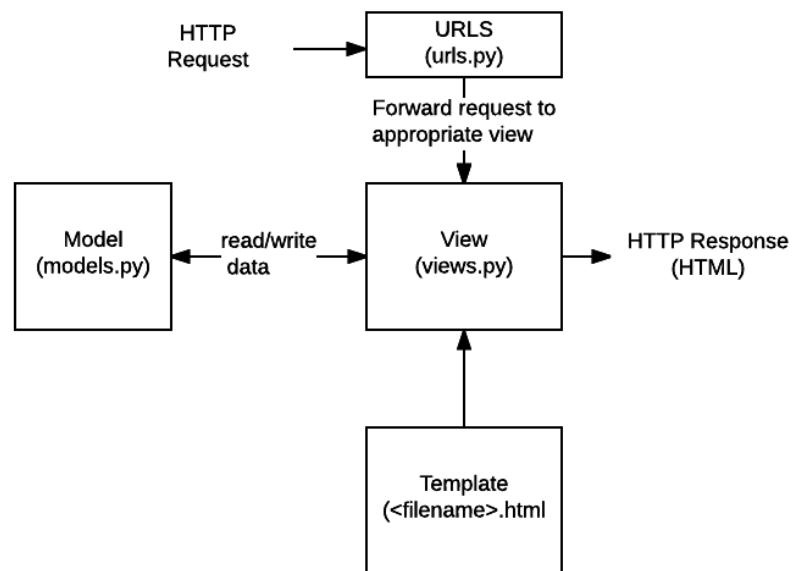


Рисунок 4.2 — Структура проекту

1. URLs: Хоча можна обробляти запити з кожного URL-адреси за допомогою однієї функції, набагато зручніше писати окрему функцію для обробки кожного ресурсу. URL-маркер використовується для перенаправлення HTTP-запитів в відповідне подання на основі URL-адреси запиту. URL-маркер також може витягувати дані з URL-адреси відповідно до заданого шаблону і передавати їх у відповідну функцію у вигляді аргументів.

2. View: Вид (view) - це функція обробника запитів, яка отримує HTTP-запити і повертає відповіді. View має доступ до даних через моделі (необхідним для задоволення запитів і делегування відповіді в шаблони).
3. Models: Моделі є об'єктами Python, які визначають структуру даних програми та надають механізми для управління (додавання, зміни, видалення) та виконання запитів до бази даних.
4. Templates: Template (шаблон) - це текстовий файл визначає структуру або розмітку сторінки (наприклад HTML сторінки), з полями для підстановки використовуваними для подання актуального вмісту. View може динамічно створювати HTML сторінки, використовуючи HTML шаблони і заповнюючи їх даними з моделі (model). Шаблон може бути використаний для визначення структури файлів будь-яких типів, не обов'язково HTML.

Будучи веб фреймворком, Django дозволяє динамічно генерувати HTML. Найпоширеніший підхід - використання шаблонів. Шаблони містять статичний HTML і динамічні дані, рендеринг яких описаний спеціальним синтаксисом.

Проект Django може використовувати один або кілька механізмів створення шаблонів (або жодного, якщо ви не використовуєте шаблони). Django надає бекенд для власної системи шаблонів, яка називається - мова шаблонів Django (Django template language, DTL), і популярного альтернативного шаблонізатора Jinja2. Сторонні додатки можуть надавати бекенд і для інших систем шаблонів.

Django надає стандартний API для завантаження і рендеринга шаблонів, незалежно від використовуваного бекенду. Завантаження включає в себе пошук шаблону за назвою і попередню обробку, зазвичай виконується завантаження шаблону в пам'ять. Візуалізація означає передачу даних контексту в шаблон і повернення рядка з результатом.

Мова шаблонів Django - власна система шаблонів Django. До Django 1.8 - це була єдина альтернатива. Це хороша система шаблонів, але зі своїми особливостями. Якщо у вас немає причин використовувати іншу систему шаблон, використовуйте вбудовану, особливо, якщо розробляєте яке розповсюджується



додаток, що містить шаблони. Вбудовані шаблони Django, які містять шаблони, наприклад `django.contrib.admin`, використовують систему шаблонів Django.

З історичних причин підтримка шаблонів і вбудована система шаблонів Django знаходяться в одному пакеті `django.template`.

### **4.3. Структура модуля авторизації користувача**

Django надає систему аутентифікації і авторизації ( "permission") користувача, реалізовану на основі фреймворку роботи з сесіями.

Система аутентифікації і авторизації дозволяє вам перевіряти повноваження користувачів і визначати які кому відповідають. Даний фреймворк включає в себе вбудовані моделі для користувачів і груп (основний спосіб застосування авторизації для більш ніж одного користувача), безпосередньо саму систему розмежування доступу (permissions) / прапори, які визначають яке завдання, з якою формою і відображенням, інструментом відображення, може працювати той, чи інший користувач.

Система аутентифікації є дуже гнучкою і дозволяє вам формувати свої власні URL-адреси, форми, відображення, а також шаблони сторінок, якщо ви побажаєте, з нуля, через простий виклик функцій відповідного API для логування користувача.

Почнемо з використання механізму Django authentication framework, щоб користувачі могли увійти на наш веб-сайт. Наше уявлення має виконувати наступні дії для входу в систему:

- Отримання username і пароля, з форми.
- Аутентифікація користувача за даними, що зберігаються в базі даних.
- Перевірка, активності користувача.
- Вхід в систему на сайті і запуск сесії.

По-перше, ми збираємося створити форму входу(рис. 4.2, рис.4.3). Ось що робить, базовий log-in view: коли уявлення `user_login` викликається із запитом GET, ми створюємо нову log-in форму з `form = LoginForm ()`, щоб відобразити його в шаблоні. Коли користувач відправляє форму за допомогою POST, ми виконуємо наступні дії:

1. Створюється екземпляр форми з відправленими даними за допомогою `form = LoginForm (request.POST)`.
2. Перевіряється валідність форми. Якщо форма невалидна, то в шаблоні відображаються помилки форми (наприклад, якщо користувач не заповнив одне з полів).
3. Якщо надіслані дані є валідними, ми шукаємо користувача в базі даних за допомогою методу `authenticate ()`. Цей метод приймає `username` і `password` і повертає об'єкт `User`. Якщо користувач не пройшов аутентифікацію, повертається необроблений `HttpResponse`, що відображається на екрані.

```
# -*- coding: utf-8 -*-
from django.shortcuts import render_to_response, redirect
from django.contrib import auth
from django.template.context_processors import csrf

def login(request):
    args = {}
    args.update(csrf(request))
    if request.POST:
        username = request.POST.get('username', '')
        password = request.POST.get('password', '')
        user = auth.authenticate(username=username, password=password)
        if user is not None:
            auth.login(request, user)
            return redirect('/')
        else:
            args['login_error'] = "Пользователь не найден"
            return render_to_response('login.html', args)
    else:
        return render_to_response('login.html', args)

def logout(request):
    auth.logout(request)
    return redirect('/')
```

Рисунок 4.2 — Представлення форми для входу

4. Якщо користувач успішно пройшов аутентифікацію, ми перевіряємо, чи активний він за допомогою атрибуту `is_active`. Якщо користувач не активний, повертається `HttpResponse`, що відображається на екрані.

5. Якщо користувач активний, він входить на сайт. Ми установлюємо сесію, викликавши метод `login()` і повертаємо повідомлення про успішне виконання.

```

{% extends 'start.html' %}

{% block login %}
  <div class="container">
    <form class="form-signin" action="/login/" method="post">
      {% csrf_token %}
      <h2 class="form-signin-heading">Введіть ім'я користувача та пароль</h2>
      <label for="username" class="sr-only">Ім'я користувача</label>
      <input type="text" name="username" id="username" class="form-control" placeholder="Ім'я користувача"
        required="" autofocus="">
      <label for="password" class="sr-only">Пароль</label>
      <input type="password" name="password" id="password" class="form-control" placeholder="Пароль" required="">
      {% if login_error %}
        <div class="alert alert-danger">{{ login_error }}</div>
      {% endif %}
      <input class="btn btn-lg btn-primary btn-block" type="submit" value="Ввійти">
    </form>
  </div>
{% endblock %}

```

Рисунок 4.3 — Шаблон форми для входу

Рисунок 4.4 — авторизація користувача

Також на цьому етапі ми створюємо захист проти міжсайтової підробки запиту. Проміжний шар CSRF і шаблонний тег надають легкий в використанні захист проти міжсайтової підробки запиту. Цей тип атак трапляється, коли зловмисний Web сайт містить посилання, кнопку форми або деякий javascript, який

призначений для виконання деяких дій на вашому Web сайті, використовуючи облікові дані авторизованого користувача, який відвідував зловмисний сайт в своєму браузері. Сюди також входить пов'язаний тип атак, 'login CSRF', де атакуємий сайт обманює браузер користувача, авторизуючись на сайті з чужими обліковими даними.

Зайшовши на стартову сторінку користувач бачить форму авторизації. Для подальшого користуванням сайту він повинен увійти в свій аккаунт. У разі некоректно введених даних, користувач побачить повідомлення про помилку (рис 4.4).

#### **4.4. Модуль пошуку товарів по заданим критеріям**

Дана система реалізує глибокий, багаторівневий пошук товарів на українському інтернет ринку.

В наш час на українському ринку роздрібної торгівлі найбільшою популярністю користується онлайн сервіс Hotline. Загалом він є дуже зручним інструментом для покупця, так як дозволяє підбирати товари по заданим характеристикам, порівнювати їх. Також на сайті можна порівнювати ціни від різних постачальників. Він підтримує майже всі інтернет магазини які функціонують на території України. Але в нього є певні недоліки. Він надає товари лише за певним обраним запитом, характеристикою, і не пропонує аналоги цих товарів. Для власника інтернет-магазину, який хоче підібрати найкращий товар для реалізації, закрити певну нішу це не зручно. Також покупця цікавить покупка найкращого товару, який може запропонувати ринок, а не за певною характеристикою. Він готовий поступитися деякими мало важливими параметрами, щоб отримати перевагу в більш важливих.

При користуванні розробленою системою, користувач вводить запит для пошуку. Запит може включати будь-які характеристики, назви, ключові слова.

Система шукає за цими запитом товари на Hotline. Можна вибрати за якою характеристикою буде виконаний цей пошук. Система знаходить товари і переходить їх сторінки. Основна проблема отримання даних в тому, що сторінка генерується динамічно, за допомогою скриптів. Тому не можна її просто завантажити по посиланню. Для вирішення цієї проблеми було використано бібліотеку Selenium WebDriver.

Selenium WebDriver - це програмна бібліотека для управління браузером. Часто вживається також більш коротку назву WebDriver.

Іноді кажуть, що це «драйвер браузера», але насправді це ціле сімейство драйверів для різних браузерів, а також набір клієнтських бібліотек на різних мовах, що дозволяють працювати з цими драйверами.

Як вже було сказано, WebDriver є сімейство драйверів для різних браузерів плюс набір клієнтських бібліотек для цих драйверів на різних мовах програмування(рис. 4.5).

В рамках проекту Selenium розробляються драйвери для браузерів Firefox, Internet Explorer і Safari, а також драйвери для мобільних браузерів Android і iOS. Драйвер для браузера Google Chrome розробляється в рамках проекту Chromium, а драйвер для браузера Opera (включаючи мобільні версії) розробляється компанією Opera Software. Тому вони формально не є частиною проекту Selenium, поширюються і підтримуються незалежно. Але логічно, звичайно, можна вважати їх частиною сімейства продуктів Selenium.

Основними поняттями в Selenium Webdriver є:

- Webdriver - найважливіша сутність, відповідальна за управління браузером. Основний хід скрипта / тесту будується саме навколо екземпляра цієї сутності.
- WebElement - друга важлива сутність, що представляє собою абстракцію над веб елементом (кнопки, посилання, інпут і ін.). WebElement

інкапсулює методи для взаємодії користувача з елементами і отримання їх текшого статусу.

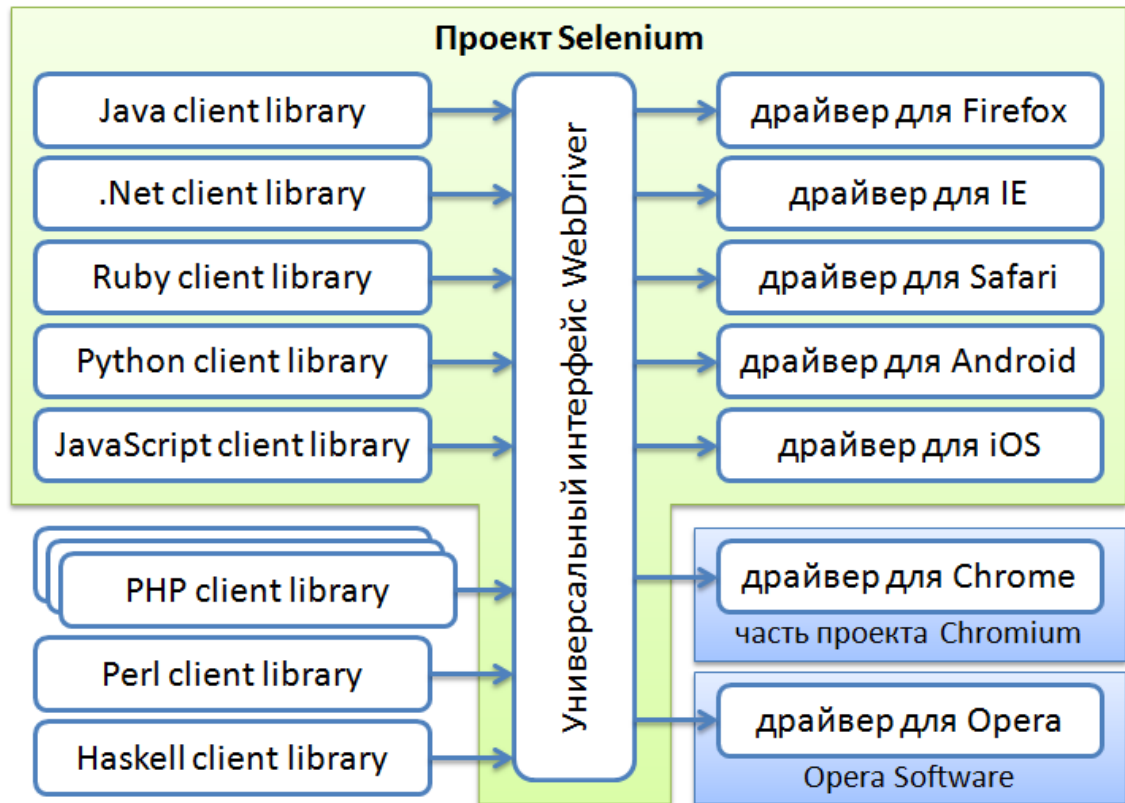


Рисунок 4.5 — Структура Selenium

- Ву - абстракція над локатором веб елемента. Цей клас інкапсулює інформацію про селекторі (наприклад, CSS), а також сам локатор елемента, тобто всю інформацію, необхідну для знаходження потрібного елемента на сторінці.

Сам процес взаємодії з браузером через Webdriver API досить простий:

- ❖ Потрібно створити Webdriver:

```
WebDriver driver = new ChromeDriver ();
```

При виконанні цієї команди буде запущений Chrome, за умови, що він встановлений в директорію за замовчуванням і шлях до ChromeDriver збережений в системній змінній PATH.

- ❖ Необхідно відкрити тестоване додаток (AUT), перейшовши по url.

В хромі при цьому повинен відкритися сайт.

❖ Далі йде серія дій по знаходженню елементів на сторінці та взаємодії з ними:

```
WebElement element = driver.findElement (By.id ( "# element_id"));
```

Отже за допомогою Selenium ми можемо емулювати сторінку, щоб на ній завантажилися всі потрібні нам дані.

На наступному етапі нам потрібно парсити сторінку, щоб витягнути з неї всі потрібні нам дані. Для цього я використовував бібліотеку BeautifulSoup.

Beautiful Soup - це бібліотека Python, яка використовує ваш попередньо встановлений синтаксичний аналізатор html / xml і перетворює веб-сторінку / html / xml в дерево, що складається з тегів, елементів, атрибутів і значень. Точніше, дерево складається з чотирьох типів об'єктів: Tag, NavigableString, BeautifulSoup і Comment. Потім це дерево можна використовувати за допомогою методів / властивостей об'єкта BeautifulSoup, створеного з бібліотеки парсерів.

Часто у вас може бути одна з наступних потреб:

1. ви можете проаналізувати веб-сторінку, щоб визначити, скільки знайдених тегів знайдено, скільки елементів кожного тега знайдено і їх значення. Можливо, ви захочете їх змінити;

1. Можливо, вам захочеться визначити імена і значення елементів, щоб ви могли використовувати їх разом з іншими бібліотеками для автоматизації веб-сторінок, таких як Selenium;

2. Можливо, ви захочете перенести / витягти дані, показані на веб-сторінці, в інші формати, такі як CSV-файл або реляційну базу даних, таку як SQLite або mysql. В цьому випадку бібліотека допоможе вам з першого кроку, зрозуміти структуру веб-сторінки, хоча ви будете використовувати інші бібліотеки для здійснення передачі;

3. Можливо, вам захочеться дізнатися, скільки елементів оформлено з використанням певного стилю CSS і які.

Послідовність для типового базового використання в вашому коді Python:

1. Імпортувати бібліотеку BeautifulSoup

2. Відкрийте веб-сторінку або html-текст за допомогою бібліотеки BeautifulSoup, вказавши, який парсер буде використовуватися. Результатом цього кроку є об'єкт BeautifulSoup. (Примітка. Зазначене ім'я парсеру має бути встановлено вже як частина ваших Python packages. Наприклад, `html.parser` - вбудований пакет, що поставляється з Python. Ви можете встановити інші синтаксичні аналізатори, такі як `lxml` або `html5lib`.)
3. «Запит» або `'object.method'` пошук об'єкта BeautifulSoup з використанням синтаксису `'object.method'` і отримати результат в колекції, наприклад, в словнику Python. Для деяких методів вихід буде простим значенням.
4. Використовуйте результат на попередньому кроці, щоб робити все, що ви хочете з ним, в решті частини вашого коду Python. Ви також можете змінити значення елементів або значення атрибутів в дереві. Модифікації не впливають на вихідний код html, але ви можете викликати методи форматування виводу (наприклад, `prettify`) для створення нового виведення з об'єкта BeautifulSoup.

Як правило, методи пошуку `.find` і `.find_all` використовуються для пошуку дерева, надаючи їм вхідних аргументів.

Вхідними аргументами є: ім'я тега, яке запитує, імена атрибутів і інші пов'язані аргументи. Ці аргументи можуть бути представлені як: рядок, регулярний вираз, список або навіть функція.

Загальне використання об'єкта BeautifulSoup включає в себе:

1. Пошук по класу CSS;
2. Пошук за адресою гіперпосилання;
3. Пошук по елементу Id, тег;
4. Пошук по імені атрибута. Значення атрибута.

Якщо вам потрібно відфільтрувати дерево за допомогою поєднання вищевказаних критеріїв, ви також можете написати функцію, яка оцінює значення `true` або `false`, і шукати по цій функції.



Отримані можливості від використання цих бібліотек дозволяють розібрати будь-яку сторінку і отримати всі потрібні нам дані.

Після виконання пошуку на першому рівні(рівень hotline), система отримує список товарів, які найбільше підходять по введеному запиту. Система зберігає ці товари в базі даних і переходить на їх сторінки. Після цього обирає найважливіші характеристики і переходить на сайти інтернет магазинів які пропонують даний товар для продажу. Виконує пошук по новим параметрам і зберігає дані про знайдену продукцію в базу даних.

Таким чином система виконує глибокий, багаторівневий уточнювальний пошук товарів за запитом.

## 4.5. Обробка результатів пошуку та відбір товарів

Після виконання пошуку і збереження товарів потрібно відібрати найбільш якісні. Якщо товар у товари багато негативних відгуків, низька оцінка при великій кількості замовлень чи коментарів, низька популярність товару або підозріла ціна (набагато менша середньої ринкової на даний товар у інших магазинів) система оцінює такий товар як неякісний( рис 4.6) і не враховує його при видачі результатів.

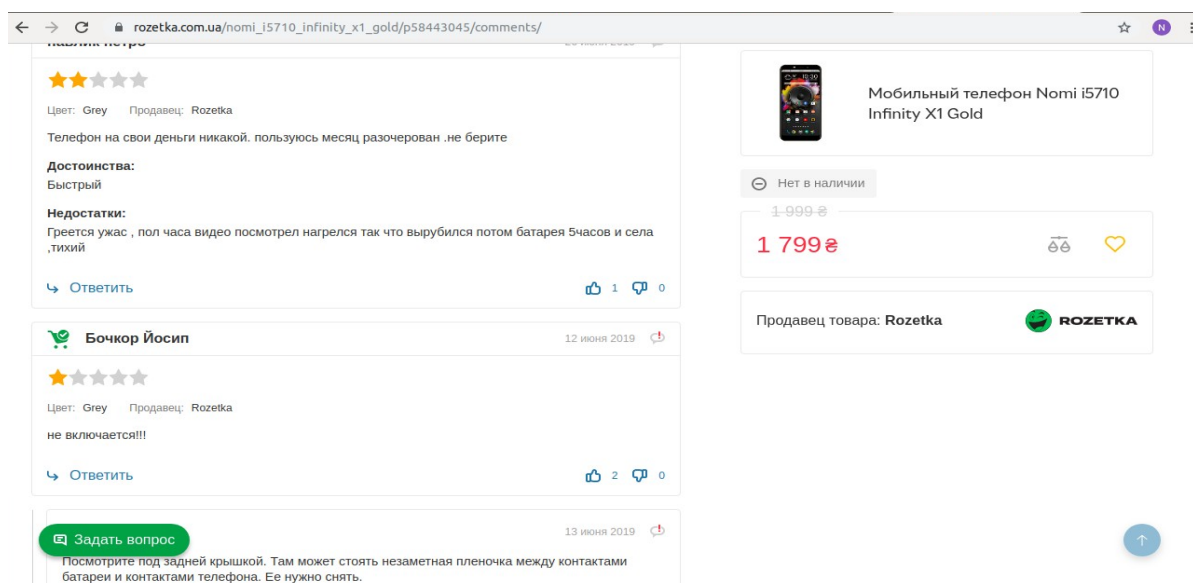


Рисунок 4.6 — Приклад неякісного товару

Такий товар не буде цікавий ні покупцям ні власникам інтернет магазинів, які проводять дослідження ринка.

Система використовує циклічний алгоритм уточнення. При виконанні глибокого пошуку може з'явитися багато результатів які дуже відрізняються від введеного користувачем запиту. Такі результати не мають своєї цінності, оскільки дуже відрізняються від очікуємого результату. Система циклічно відсіює такі товари.

#### **4.6. Модуль візуалізація інформації**

Після роботи попередніх модулів ми маємо набір даних, який потрібно візуалізувати в зручному для користувача вигляді. Для цього було обрано бібліотеку Matplotlib.

Бібліотека matplotlib - це бібліотека двовимірної графіки для мови програмування python за допомогою якої можна створювати високоякісні малюнки різних форматів. Matplotlib являє собою модуль-пакет для python.

Переваги Matplotlib:

- різні типи графіків (діаграми, точкові, лінійні, контурні ...);
- синтаксис на базі Python;
- інтеграція з науковим обчислювальним пакетом NumPy;
- вихідні дані можуть бути списками і кортежами Python, або масивами NumPy;
- настраюється формат графіка (масштаби осей, підписи, сітка і т.д.);
- настраюється текст (тип шрифту, розмір, накреслення і т.д.);
- форматування TeX (рівняння, грецькі символи і т.д.);

- сумісний з IPython (дозволяє будувати графіки інтерактивно з командної оболонки Python);
- автоматизація - використовує цикли Python для ітеративного побудови графіків;
- збереження графіків у вигляді зображень (в форматах png, pdf, ps, eps і svg).

Основою ефективної роботи matplotlib і багатьох його особливостей служить синтаксис на базі Python. Є багато наукових пакетів, які дозволяють будувати якісні графіки, але багато хто з них дозволяють робити це безпосередньо в вашому коді на Python? Крім того, можуть вони створювати автоматизовані процедури для ітеративного побудови графіків і збереження їх в графічних файлах? Matplotlib дозволяє виконувати всі ці завдання. З його допомогою ви можете заощадити час, що витрачають на створення графіків вручну.

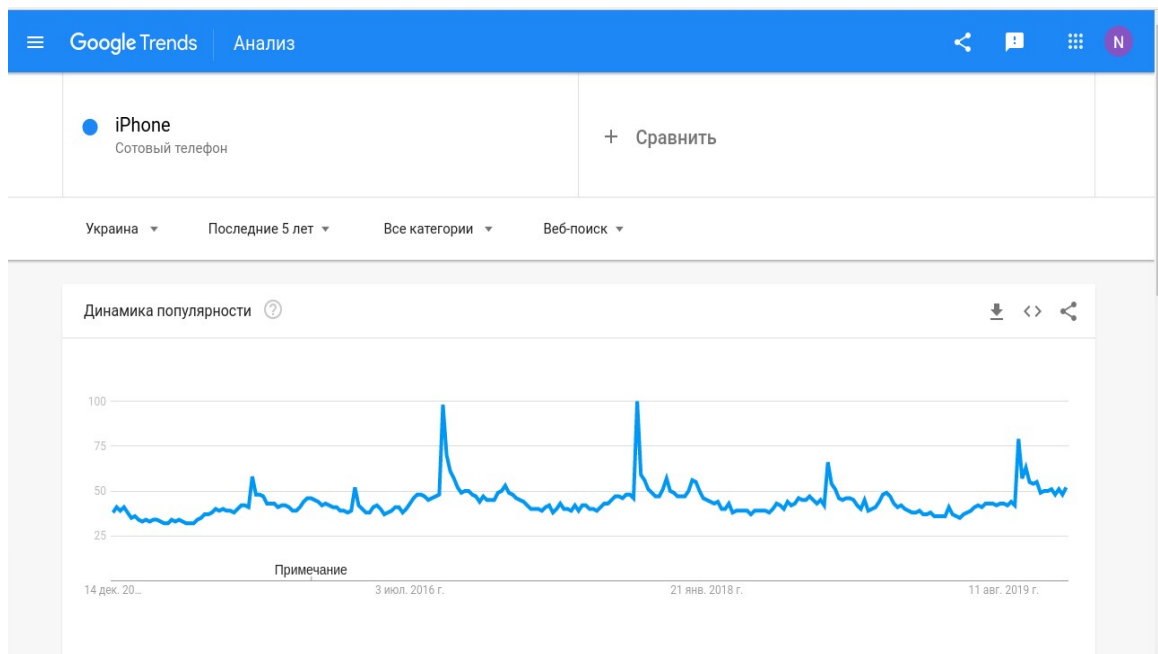


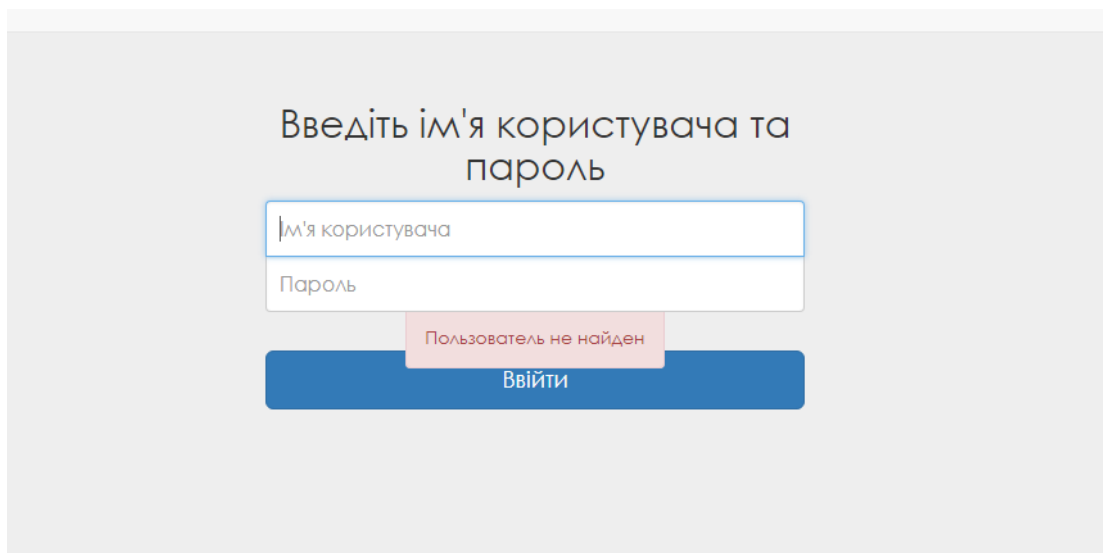
Рисунок 4.6 — приклад роботи google trends

Ця бібліотека дає можливість створені графіки зберігати в базі даних і передавати в html файл. Тобто дозволяє не використовувати javascript для візуалізації графіків.

Також для зручності проведення аналізу ринку було використано сервіс google trends. Його можна вмонтувати на будь-яку сторінку, попередньо передавши параметри товару. Загалом він дає змогу зрозуміти як змінювалась тенденція пошуку цього товару, наскільки він цікавить потенційних покупців.

## 5. МЕТОДИКА РОБОТИ КОРИСТУВАЧА ІЗ ПОШУКОВО-АНАЛІТИЧНОЮ СИСТЕМОЮ ПІДТРИМКИ ПРОЦЕСІВ АНАЛІЗУ ТОВАРНОГО РИНКУ

При переході на розроблений веб-сайт користувач повинен ввести логін і пароль для авторизації(рис 5.1).



Введіть ім'я користувача та  
пароль

Ім'я користувача

Пароль

Пользователь не найден

Ввійти

Рисунок 5.1 — авторизація користувача

Після успішного введення даних він отримає доступ до користуванні ресурсом.

На наступній сторінці йому пропонують ввести запит для пошуку. Цей запит може містити в собі будь які характеристики, назви або типи товарів (рис.5.2). Також тут він може вказати за яким параметром сортувати знайдений товар. Після натиснення на кнопку ОК система почне свою роботу.

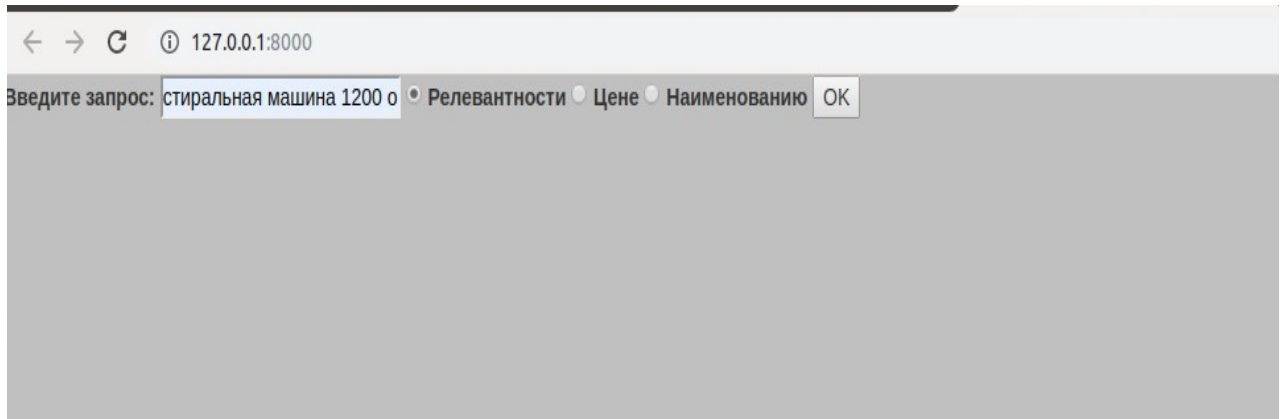


Рисунок 5.2 — введення запиту

В результаті користувач отримує сторінку з списком знайдених товарів(рис 5.3). При бажанні він може натиснути на кнопку “Таблиця” та отримати ці дані у вигляді таблиці, для порівняння характеристик(рис. 5.4).

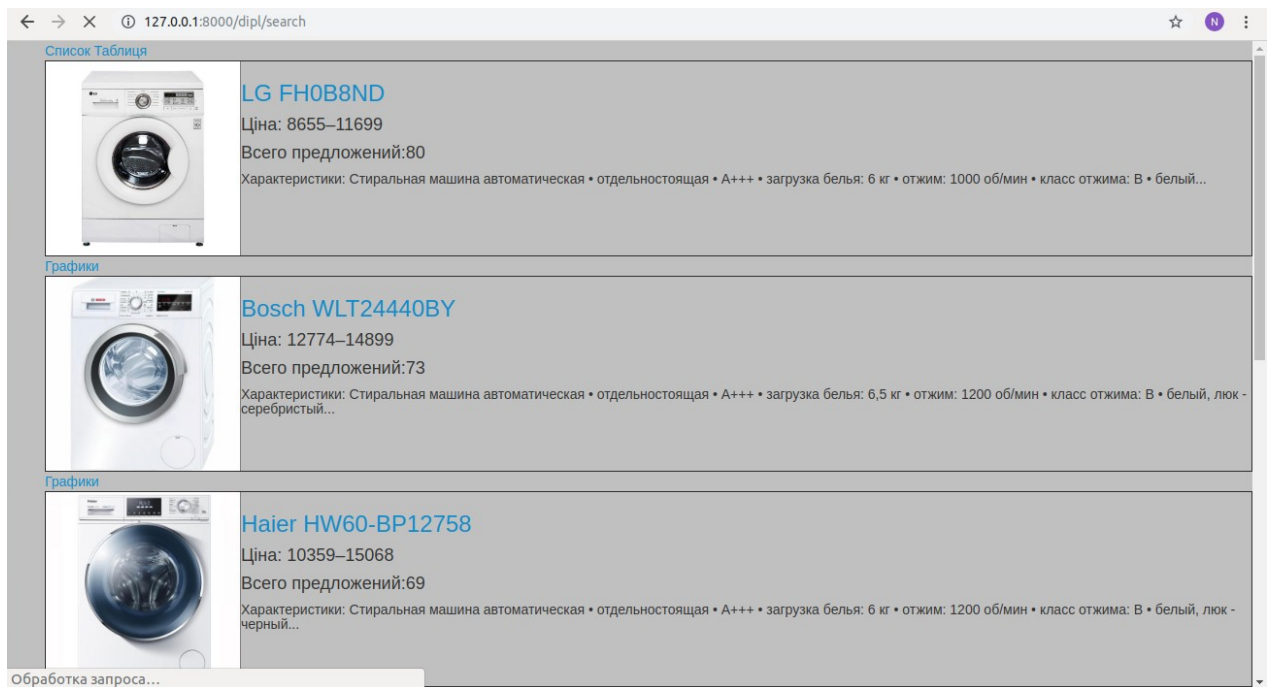


Рисунок 5.3 — отримання результатів пошуку у вигляді списку

При бажанні користувач може натиснути на кнопку “Графіки” та отримати за кожним з знайдених товарів графіки популярності запиту на цей товар, динаміку

зміни ціни в гривнях(рис 5.5), доларах, кількість поставщиків, та попит на цей товар(рис 5.6).

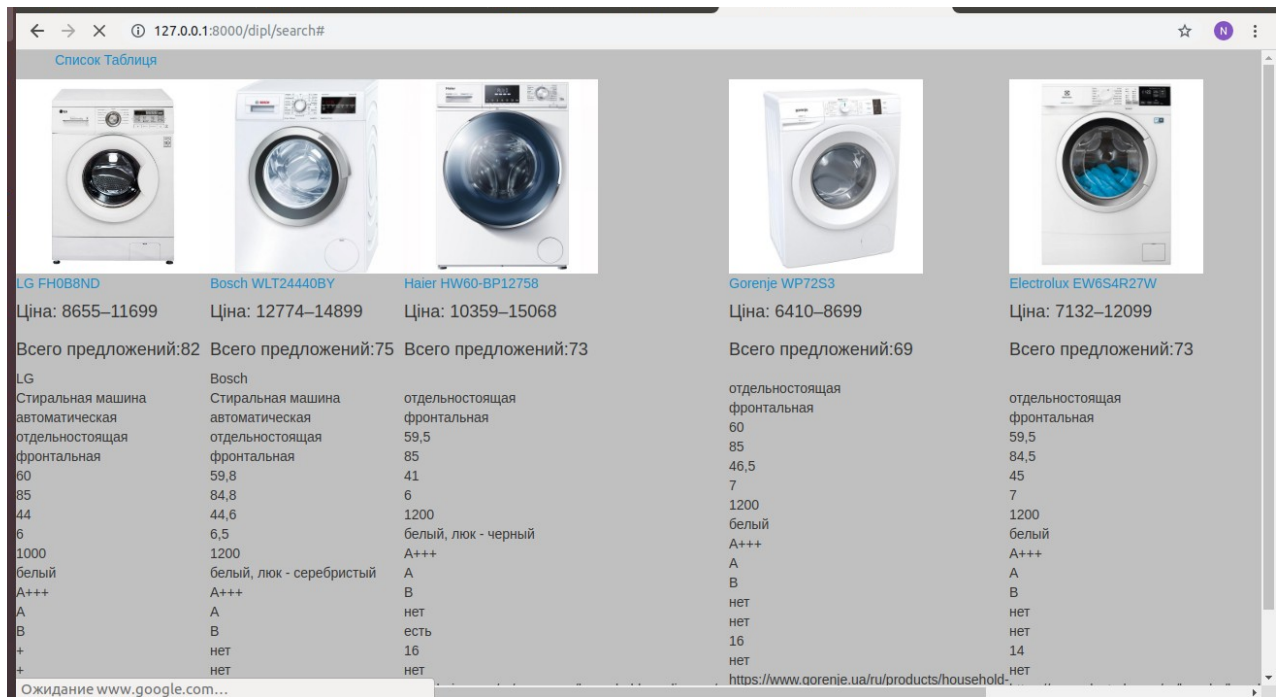


Рисунок 5.4 — отримання результатів пошуку у вигляді таблиці

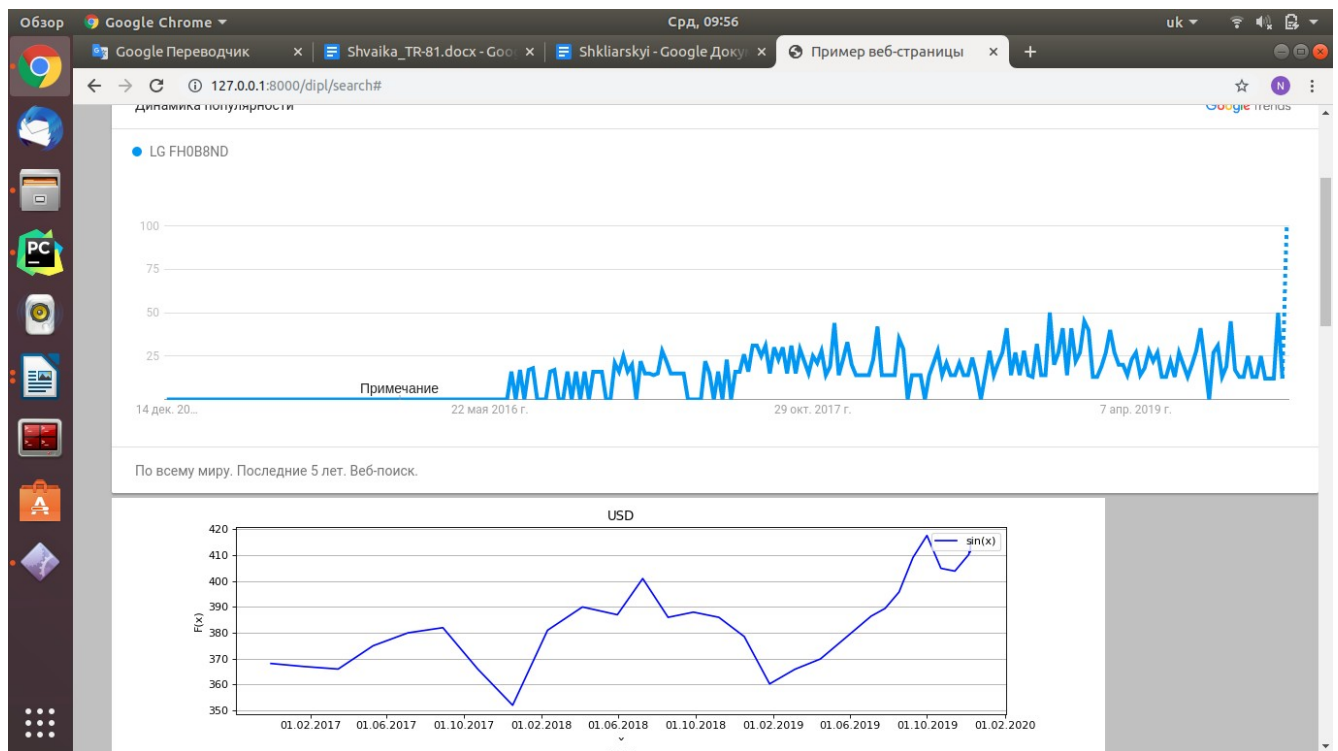


Рисунок 5.5 — графіки google trends та ціна в доларах



Рисунок 5.6 — графіки кількості постачальників та попит



## 6. РОЗРОБКА СТАРТАП-ПРОЕКТУ

В даному розділі викладено підхід для розробки рішення, яке реалізує викладене в роботі напрацювання. Провівши аналіз ринку були описані цілі та ідея, сформульовані основні вимоги, визначені сильні та слабкі сторони потенційного комерційного продукту в результаті SWOT-аналізу.

### 6.1. Опис ідеї проекту

ПЗ є веб застосунком та може бути використане для пошуку товару та аналізу ринку.

У якості найближчого аналогу прийнято сервіс Hotline. Сервіс дозволяє підбирати товари та аналізувати попит.

До недоліків найближчого аналога відноситься вузьконаправлений пошук та сумнівна якість підібраних товарів.

В основу розробки ПЗ поставлено задачу удосконалення функціоналу вже існуючих на ринку конкурентів, шляхом покращення пошуку та відбору найкращих товарів..

Розглянемо зміст ідеї, можливі напрямки застосування, основні переваги, які зможе отримати користувач представлено у таблиці 6.1.

Таблиця 6.1 – Опис ідеї стартап-проекту.

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Удосконалення програмного забезпечення для пошуково-аналітичної системи підтримки процесів аналізу товарного ринку	1.Приватні	Вдосконалений та точніший пошук
	2. Комерційні	Аналіз товарного ринку.

В результаті дослідження сильних, слабких та нейтральних характеристик ідеї проекту було проаналізовано техніко-економічні характеристики існуючих на ринку конкурентів, що, у свою чергу, надає можливість спрогнозувати частку майбутніх потенційних користувачів. Тож, запропоноване ПЗ вирізняється конкурентоспроможним потенціалом серед перелічених вище характеристик, адже містить близько 8% в якості показників слабкої сторони, 51% - нейтральної сторони, 41% - сильної сторони. А, враховуючи, що основна мета майбутнього ПЗ – удосконалення вже існуючих систем з подібним функціоналом, маємо досить успішні наміри завоювати прихильність на ринку.

## 6.2. Технологічний аудит ідеї проекту

Далі наведений аудит технології, за допомогою якої можна реалізувати ідею проекту. Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 6.2):

Таблиця 6.2 – Технологічна здійсненність ідеї проекту.

№ п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1.	Пошуково-аналітична система підтримки процесів аналізу товарного ринку	Python в середовищі Pycharm.	Наявні	Доступні для вільного користування
2.	Створення графічного інтерфейсу програмного додатку.	Фреймворк Django, фреймворк Bootstrap	Наявні	Доступні для вільного користування
4.	Парсинг інтернет магазинів	Мова програмування Python, бібліотеки selenium та beautifulsoup	Наявні	Доступні для вільного користування
Обрана технологія реалізації ідеї проекту: веб додаток планується розробити мовою програмування Python в середовищі Pycharm з використанням Django. Для можливості парсинг інтернет магазинів				

бібліотеки selenium та beautifulsoup , для можливості графічного відображення – бібліотека “matplotlib”.

### 6.3. Аналіз ринкових можливостей запуску стартап-проекту

Зробимо аналіз попиту. Далі в таблиці 6.3 наводиться його аналіз, виходячи з наявного ринку.

Таблиця 6.3 – Попередня характеристика потенційного ринку стартап-проекту.

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	3 головних гравців
2.	Загальний обсяг продаж, грн/ум.од	\$10 млрд
3.	Динаміка ринку (якісна оцінка)	зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	затвердження ліцензійних умов провадження, створення маркетингової стратегії для проведення ефективної рекламної діяльності стосовно ПЗ
5.	Специфічні вимоги до стандартизації та сертифікації	відсутні
6.	Середня норма рентабельності в галузі (або по ринку), %	+15,9%

З таблиці 6.3 можна зробити висновок, що ринок є привабливим для входження за попереднім оцінюванням.

В таблиці 6.4 наведено основні характеристики потенційних клієнтів стартап-проекту.

Таблиця 6.4 – Характеристика потенційних клієнтів стартап-проекту.

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Аналіз ринку.	Власники інтернет	-	Підбір найкращих

		магазинів		товарів по заданим характеристикам , аналіз товарів.
--	--	-----------	--	--

З таблиці 6.4 можна зробити висновок, що цільовою аудиторією є інтернет магазини, які займаються роздрібною торгівлею. Для всіх аудиторій важлива якість продукції та технічні характеристики.

Проведемо аналіз ринкового середовища та сформулюємо фактори загроз та фактори можливостей у таблицях 6.5 та 6.6.

Таблиця 6.5 – Фактори загроз.

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Недосвідчені учасники команди	Призначення недосвідчених працівників (студентів) для виконання роботи проекту ставить під загрозу дату його завершення, оскільки їм може знадобитися більше часу, щоб ознайомитися з бізнес-моделлю, технологіями.	Щоб мінімізувати цей ризик, необхідно закласти достатньо часу на введення нових працівників у курс справи.
2.	Планування та послідовність виконання задач	Навіть якщо ці завдання виконують різні люди, їх одночасне виконання у великій кількості створює ризик для проекту, особливо наприкінці його реалізації.	Перевірити, чи не заплановано забагато завдань на один і той самий час. При плануванні задач проекту спочатку необхідно скласти список завдань і згрупувати їх, щоб оцінити весь обсяг проекту та кінцеві результати. Потім можна починати зв'язувати завдання,

			щоб отримати ідеальний розклад.
3.	Потужна клієнтська база конкурентів	Конкуренти, які мають впевнений досвід продукта на ринку здобули сильну базу клієнтів- споживачів.	Розвиток вражаючої маркетингової кампанії, створення стратегії піар- менеджменту, закладання регламенту рекламної кампанії, акційних пропозицій.

Таблиця 6.6 – Фактори можливостей.

Заключним етапом ринкового аналізу, щодо інтеграції продукту є SWOT-аналіз. SWOT-аналіз (або SWOT-матриця) - це метод стратегічного планування, який допомагає людині чи організації визначити сильні сторони, слабкі сторони, можливості та загрози, пов'язані з діловими конкурентами чи плануванням проекту. Даний аналіз має на меті визначити цілі підприємницької діяльності або проекту та визначити внутрішні та зовнішні фактори, які є сприятливими та несприятливими для досягнення цих цілей. Користувачі аналізу SWOT часто запитують та відповідають на запитання, щоб створити значущу інформацію для кожної категорії, щоб інструмент був корисним та визначити їх конкурентну перевагу. SWOT був описаний як спрощений і справжній інструмент стратегічного аналізу.

Назва є аббревіатурою для чотирьох параметрів, які розглядає техніка.

Сильні сторони: характеристики бізнесу або проекту, що дають йому перевагу перед іншими.

Слабкі сторони: характеристики бізнесу, що ставлять бізнес або проект у не вигідному становищі порівняно з іншими.

Можливості: елементи у навколишньому середовищі, які бізнес або проект можуть використати для його переваг.

Загрози: елементи в середовищі, які можуть спричинити проблеми для бізнесу чи проекту.

SWOT – матриця програмно-апаратного комплексу представлена у таблиці 6.7.

З SWOT- аналізу стартап-проекту можна зробити висновок, що сильними сторонами є цілодобова підтримка, інструкція по експлуатації, якість продукту, відповідність потребам споживачів та доступність. А слабкими сторонами є низька репутація компанії на початку впровадження проекту в життя та присутність багів.

Таблиця 6.7 – SWOT- аналіз стартап-проекту

<p>Сильні сторони:</p> <p>Цілодобова технічна підтримка; справка-інструкція по експлуатації; якість продукту; продукт відповідає потребам споживачів; доступність.</p>	<p>Слабкі сторони:</p> <p>Низька репутація компанії на початку впровадження проекту в життя; присутність багів.</p>
<p>Можливості:</p> <p>Вихід на міжнародний ринок; результативність; розвиток нових функціональних можливостей.</p>	<p>Загрози:</p> <p>Зниження доходів потенційних клієнтів; блокування реклами на просторах інтернету, соціальних мереж; блокування інтернет-ресурсу програмного забезпечення.</p>

## ВИСНОВКИ

Розроблено пошуково-аналітична система підтримки процесів аналізу товарного ринку.

Даний продукт написаний на мові Python з використанням Django. Завдяки цьому, вдалося досягти поставлених цілей, таких як легка підтримка системи, можливість її масштабування, кросплатформність, доповнення додатковим функціоналом і модулями. В даній програмі використовувались новітні можливості мови Python.

Даний додаток може входити до різних систем, які призначені для пошуку, аналізу та продажу товару. Розроблений програмний продукт може використовуватися у програмних системах для різних платформ.

Розроблений програмний продукт дозволяє проводити глибокий пошук, витяг інформації та за допомогою циклічного алгоритма уточнення обирати найкращі з представлених варіантів.

Результати даної роботи можуть бути використані власниками інтернет магазинів як для створення магазину з нуля, так і для підтримки уже існуючих. Також системою можуть користуватися для підбору товару звичайні користувачі.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1.                   Електронная       коммерция в Украине и в мире – в цифрах.       –  
[Электронный ресурс]. – Режим доступа :       <[https://www.web-mashina.com/web-  
blog/ecommerce-prognoz-elektronnoi-kommercii-ukrainy-2017-2018](https://www.web-mashina.com/web-blog/ecommerce-prognoz-elektronnoi-kommercii-ukrainy-2017-2018)>. Дата доступа:  
28.11.2019.
  
2.                   Лутц,       М. Программирование на Python, II том / М.       Лутц. - СПб.:  
Символ-плюс, 2015. - 992 с.
3.                   Лутц,       М. Программирование на Python, I том / М.       Лутц. - СПб.:  
Символ-плюс, 2015. - 992 с.
4.                   Абрютина М. С. Экономический анализ товарного       рынка и торговой  
деятельности / Абрютина       М. С. Дис, 2010.- 464 с.
5.       Django .       – [Электронный ресурс]. – Режим доступа       :  
      <<https://www.djangoproject.com/>>. Дата доступа: 27.11.2019.
6.       Design Patterns, Elements of Reusable       Object-Oriented Software / E.Gamma,  
R. Helm, R. Johnson, J. Vlissides. —       Boston: Addison–Wesley, 1994. —       395 с.
7.       Джонс А. Программирование       в сетях Microsoft Windows / А. Джонс, Дж.  
Оланд.       —       Москва: Питер, 2002. —       601 с.
8.       Стивенс Р. Алгоритмы. Теория и практическое применение / Род Стивенс. —  
Москва: Эксмо, 2017. — 544 с.
9.       Скиена С. Алгоритмы. Руководство по разработке / Стивен       Скиена. —  
Санкт-Петербург: БХВ-Петербург, 2017. —       720 с.
10.       Джонс Э. Программирование       в сетях Microsoft Windows / Э. Джонс, Д.  
Оланд. —       Москва: Питер, 2002. —       594 с.
11.       Форум програмістів       [Електронний ресурс] —       Режим доступу до  
ресурсу:       <https://stackoverflow.com/>

12. Довідник по шаблонам проектування [Електронний ресурс] — Режим доступу до ресурсу: <http://cpp-reference.ru/patterns/>
13. Python [Електронний ресурс] — Режим доступу до ресурсу: <https://docs.python.org/3/>
14. Beautiful Soup [Електронний ресурс] — Режим доступу до ресурсу: <https://www.crummy.com/software/BeautifulSoup/>
15. Selenium [Електронний ресурс] — Режим доступу до ресурсу: <https://selenium.dev/>
16. Matplotlib [Електронний ресурс] — Режим доступу до ресурсу: <https://matplotlib.org/>
17. Django на русском [Електронний ресурс] — Режим доступу до ресурсу: <https://djbook.ru/>
18. Крокфорд Д. JavaScript. Сильные стороны — Москва: Питер, 2012. — 354 с.
19. Hypertext Transfer Protocol overview. — Режим доступу: <https://www.w3.org/Protocols/>. — Дата доступу: 18.09.2019.
20. Design Patterns, Elements of Reusable Object-Oriented Software / E.Gamma, R. Helm, R. Johnson, J. Vlissides. — Boston: Addison–Wesley, 1994. — 395 с.



**Кузьмініх В. О., к.т.н., доцент**

*Національний технічний університет України «Київський політехнічний інститут імені*

*Ігоря Сікорського», м. Київ*

*АПЕПС, доцент*

**Шклярський Н.О**

*Національний технічний університет України «Київський політехнічний інститут*

*імені Ігоря Сікорського», м. Київ*

*АПЕПС, студент*

## **ПОШУКОВО – АНАЛІТИЧНА СИСТЕМА ПІДТРИМКИ ПРОЦЕСІВ АНАЛІЗУ ТОВАРНОГО РИНКУ**

В 2018 порівняно з попереднім ринок електронної комерції виріс на третину. Українці купили товарів і послуг на суму 65 млрд грн. Найчастіше аналіз ринку допомагає визначити нішу, яка поки що не надто зайнята та де можна отримати великий попит і дозволить оцінити попит з боку покупців, і вже на підставі отриманої інформації приймати відповідне рішення. Тому аналіз потрібно проводити ще на тому етапі, коли підприємець вибирає товар, який буде продавати в Інтернет-магазині.

Розроблена система реалізує глибокий, багаторівневий пошук товарів на українському інтернет ринку. В наш час на українському ринку роздрібної торгівлі найбільшою популярністю користується онлайн сервіс Hotline. Загалом він є дуже зручним інструментом для покупця, так як дозволяє підбирати товари по заданим характеристикам, порівнювати їх. Також на сайті можна порівнювати ціни від різних постачальників. Він підтримує майже всі інтернет магазини які функціонують на території

України. Але в нього є певні недоліки. Він надає товари лише за певним обраним запитом, характеристикою, і не пропонує аналоги цих товарів. Для власника інтернет-магазину, який хоче підібрати найкращий товар для реалізації, закрити певну нішу це не зручно. Також покупця цікавить покупка найкращого товару, який може запропонувати ринок, а не за певною характеристикою. Він готовий поступитися деякими мало важливими параметрами, щоб отримати перевагу в більш важливих.

При користуванні розробленою системою, користувач вводить запит для пошуку. Запит може включати будь-які характеристики, назви, ключові слова. Система шукає за цими запитом товари на Hotline. Після виконання пошуку на першому рівні(рівень hotline), система отримує список товарів, які найбільше підходять по введеному запиті. Система зберігає ці товари в базі даних і переходить на їх сторінки. Після цього обирає найважливіші характеристики і переходить на сайти інтернет магазинів які пропонують даний товар для продажу. Виконує пошук по новим параметрам і зберігає дані про знайдену продукцію в базу даних.

Таким чином система виконує глибокий, багаторівневий уточнювальний пошук товарів за запитом. Після виконання пошуку і збереження товарів потрібно відібрати найбільш якісні. Якщо товар у товару багато негативних відгуків, низька оцінка при великій кількості замовлень чи коментарів, низька популярність товару або підозріла ціна (набагато менша середньої ринкової на даний товар у інших магазинів) система оцінює такий товар як неякісний і не враховує його при видачі результатів. Такий товар не буде цікавий ні покупцям ні власникам інтернет магазинів, які проводять дослідження ринка.

Система використовує циклічний алгоритм уточнення. При виконанні глибокого пошуку може з'явитися багато результатів які дуже відрізняються

від введеного користувачем запити. Такі результати не мають своєї цінності, оскільки дуже відрізняються від очікуємого результату. Система циклічно відсіює такі товари.

Перелік посилань:

1. Электронная коммерция в Украине и в мире – в цифрах. – [Электронный ресурс]. – Режим доступа : <<https://www.web-mashina.com/web-blog/ecommerce-prognoz-elektronnoi-kommercii-ukrainy-2017-2018>>.